



UNIVERSIDAD CARLOS III DE MADRID
Ingeniería Informática

DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN MÓVIL PARA EL CONTROL DE UN SISTEMA DOMÓTICO

PROYECTO DE FIN DE CARRERA
Madrid, junio 2017

Autor: Imanol Sanz Ruiz.
Tutor: Javier Fernández Muñoz

Título: Diseño e implementación de una aplicación móvil para el control de un sistema domótico

Autor: Imanol Sanz Ruiz

Director: Javier Fernández Muñoz

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Grado el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACION de

VOCAL

SECRETARIO

PRESIDENTE

A mi familia y amigos.

*“Any tree can drop an apple.
I’ll drop the freaking moon”
Sho Minamimoto*

RESUMEN

El documento actual recoge los datos generados por el desarrollo de un sistema domótico controlado con una aplicación móvil Android. Muestra cómo se forman los módulos en los que se divide el proyecto y como estos se implementan posteriormente en el sistema para soportar la funcionalidad necesaria para cada usuario. La aplicación mencionada, podrá enviar pedidos y recibir datos del sistema domótico, con el fin de controlar el entorno alrededor de la puerta de la casa de los usuarios.

El documento sigue el proceso de desarrollo que hemos realizado para crear el sistema, comienza listando los objetivos y motivaciones que han dado paso a este proyecto, que es evidentemente el primer paso para el proyecto, y que resulta fundamental para facilitar el resto. Continuamos analizando lo que ya hay disponible en el mercado, y las diferentes opciones con las que contamos para desarrollar nuestro producto. Una vez hecho esto, pasamos a analizar nuestro sistema y a diseñarlo, este paso es fundamental, dado que si se realiza correctamente la implementación posterior del sistema es mucho más sencilla. Terminado el diseño, comenzamos con la implementación y las pruebas de la misma. Lo siguiente que hacemos es exponer la planificación que hemos seguido y los gastos que ha generado el proyecto. Y para acabar, las conclusiones que hemos alcanzado durante el desarrollo completo de este proyecto de fin de grado.

Abstract

The current document collects all the data generated from developing a home automation system controlled with a Android mobile phone application. It shows how the different modules that this project is divided in were designed, and later on implemented into the system in order to support the functionality needed for each user. The application mentioned, will be able to send orders and receive data from the home automation system, in order to control the enviroment around the users house door.

The document follows the development process that we have done to create the system. It begins by listing the objectives and motivations that have given way to this project, which is obviously the first step for the project, which is fundamental to facilitate the rest. We continue to analyze what is already available in the market, and the different options we must develop our product. Once this is done, we proceed to analyze our system and design it, this step is fundamental, since if it is done correctly the later implementation of the system is much easier. Finished the design, we started with the implementation and testing of it. The next thing we do is explain the planning that we have followed and the expenses generated by the project. And to finish, the conclusions that we have reached during the complete development of this project of end of degree.

Índice de contenidos

Chapter 1: Introduction and objectives	20
1.1 General Vision	21
1.2 Motivation.....	21
1.3 Objectives.....	22
1.4 Development phases.....	23
1.5 Document's structure	23
Capítulo 2: Entorno tecnológico	26
2.1 Evolución de la tecnología.....	27
2.2 sistemas domóticos más utilizados actualmente.....	28
2.3 sistemas domóticos más utilizados en el ámbito de seguridad.	34
2.4 Análisis de la plataforma móvil	39
Criterios de elección.....	40
2.5 Análisis de hardware	41
2.5.1 Análisis de la placa para soportar el sistema domótico.	41
2.5.2 Sensores:	42
2.5.3 Actuadores:	42
2.6 Tecnología utilizada.....	42

Capítulo 3: Análisis diseño del sistema	43
3.1: Análisis.....	44
3.1.1: Requisitos de usuario	44
Requisitos de capacidad:.....	44
Requisitos de restricción:	47
3.1.2: Casos de uso.....	48
3.1.3: Requisitos software.....	54
Requisitos funcionales:	54
Requisitos no funcionales:	58
3.1.4: Matriz de trazabilidad	59
3.2: Diseño.....	60
3.2.1: Diseño de la arquitectura.....	60
3.2.2: Diseño de la interfaz.....	65
3.2.3: Diagrama de navegación.....	71
3.3 diseño hardware	72
Capítulo 4: Implementación y pruebas	74
4.1 implementación	75
4.1.1 implementación de la aplicación móvil.....	75
4.1.2 implementación del sistema domótico.....	76
4.2 Sistema de tiempo real	77
Encendido/apagado de las luces.....	78
Apertura/cierre de la puerta.....	79
Tomar imagen con la cámara.....	79
Lectura del sensor de movimiento.....	79
Lectura del sensor de luz.....	80
Comunicación con la aplicación móvil	80
Planificador cíclico.....	80
Implementación del planificador	82
4.3 Pruebas.....	83
Capítulo 5: Planificación y presupuesto	90
5.1 Planificación	91
5.1.1 Diagrama de Gantt	91
5.2 Presupuesto	92
5.2.1 tiempo dedicado	92
5.2.2 Coste personal.....	92

5.2.3 Coste de hardware	93
5.2.4 Coste de software	94
5.2.5 Costes totales	95
Chapter 6: Conclusions and future lines	96
6.1 Conclusions.....	97
6.1.1 Project conclusions.....	97
6.1.2 Personal conclusions	98
6.2 Future working lines.....	99
Anexo 1: Glosario	101
Anexo 2: Bibliografía	102

Índice de tablas

<i>Tabla 1: diferencia entre Arduino y Raspberry</i>	<i>41</i>
<i>Tabla 2: RU_CA_01</i>	<i>44</i>
<i>Tabla 3: RU_CA_02</i>	<i>44</i>
<i>Tabla 4: RU_CA_03</i>	<i>45</i>
<i>Tabla 5: RU_CA_04</i>	<i>45</i>
<i>Tabla 6: RU_CA_05</i>	<i>45</i>
<i>Tabla 7: RU_CA_06</i>	<i>45</i>
<i>Tabla 8: RU_CA_07</i>	<i>45</i>
<i>Tabla 9: RU_CA_08</i>	<i>45</i>
<i>Tabla 10: RU_CA_09</i>	<i>46</i>
<i>Tabla 11: RU_CA_10</i>	<i>46</i>
<i>Tabla 12: RU_CA_11</i>	<i>46</i>
<i>Tabla 13: RU_CA_12</i>	<i>46</i>
<i>Tabla 14: RU_CA_13</i>	<i>46</i>
<i>Tabla 15: RU_CA_14</i>	<i>46</i>
<i>Tabla 16: RU_CA_15</i>	<i>47</i>
<i>Tabla 17: RU_CA_16</i>	<i>47</i>
<i>Tabla 18: RU_RE_01</i>	<i>47</i>
<i>Tabla 19: RU_RE_02</i>	<i>47</i>
<i>Tabla 20: RU_RE_03</i>	<i>47</i>
<i>Tabla 21: CU_01</i>	<i>49</i>
<i>Tabla 22: CU_02</i>	<i>50</i>
<i>Tabla 23: CU_03</i>	<i>50</i>
<i>Tabla 24: CU_04</i>	<i>51</i>
<i>Tabla 25: CU_05</i>	<i>51</i>
<i>Tabla 26: CU_06</i>	<i>52</i>
<i>Tabla 27: CU_07</i>	<i>52</i>
<i>Tabla 28: CU_08</i>	<i>53</i>
<i>Tabla 29: CU_09</i>	<i>53</i>
<i>Tabla 30: RS_FU_01</i>	<i>54</i>
<i>Tabla 31: RS_FU_02</i>	<i>54</i>
<i>Tabla 32: RS_FU_03</i>	<i>55</i>
<i>Tabla 33: RS_FU_04</i>	<i>55</i>

Tabla 34: RS_FU_05.....	55
Tabla 35: RS_FU_06.....	55
Tabla 36: RS_FU_07.....	55
Tabla 37: RS_FU_08.....	55
Tabla 38: RS_FU_09.....	56
Tabla 39: RS_FU_10.....	56
Tabla 40: RS_FU_11.....	56
Tabla 41: RS_FU_12.....	56
Tabla 42: RS_FU_13.....	56
Tabla 43: RS_FU_14.....	56
Tabla 44: RS_FU_15.....	57
Tabla 45: RS_FU_16.....	57
Tabla 46: RS_FU_17.....	57
Tabla 47: RS_FU_18.....	57
Tabla 48: RS_FU_19.....	57
Tabla 49: RS_FU_20.....	57
Tabla 50: RS_FU_21.....	58
Tabla 51: RS_FU_22.....	58
Tabla 52: RS_NF_01.....	58
Tabla 53: RS_NF_02.....	58
Tabla 54: RS_NF_03.....	58
Tabla 55: Matriz de trazabilidad [1/2]	59
Tabla 56: Matriz de trazabilidad [2/2]	60
Tabla 57: CO_01 Gestor de luces	63
Tabla 58: CO_02 Visor de luces	63
Tabla 59: CO_03 Gestor de puerta	63
Tabla 60: CO_04 Visor de puerta	64
Tabla 61: CO_05 Gestor de cámara	64
Tabla 62: CO_06 Visor de cámara	64
Tabla 63: CO_07Gestor de comunicación.....	64
Tabla 64: CO_08 Controlador	65
Tabla 65: CO_09 Gestor de sensores	65
Tabla 66: periodos y tiempos de computo.....	80
Tabla 67: ciclos secundarios de las tareas.....	81
Tabla 68: configuración valida de los ciclos secundarios.....	81
Tabla 69: PR_01.....	84
Tabla 70: PR_02.....	84
Tabla 71: PR_03.....	84
Tabla 72: PR_04.....	85
Tabla 73: PR_05.....	85
Tabla 74: PR_06.....	85
Tabla 75: PR_07.....	86
Tabla 76: PR_08.....	86
Tabla 77: PR_09.....	86
Tabla 78: PR_10.....	87
Tabla 79: PR_11.....	87
Tabla 80: PR_12.....	87
Tabla 81: PR_13.....	88
Tabla 82: PR_14.....	88
Tabla 83: PR_1.....	88
Tabla 84: PR_16.....	89
Tabla 85: PR_17.....	89

Tabla 86: planificación del proyecto..... 91

Tabla 87: costes de personal 93

Tabla 88: Costes de hardware 94

Tabla 89: Costes de Software 94

Tabla 90: Costes totales..... 95

Índice de imágenes

<i>imagen 1: general idea of the system.</i>	21
<i>imagen 2: ejemplo de sistema domótico</i>	27
<i>imagen 3: uso histórico de la telefonía móvil</i>	28
<i>imagen 4: Amazon Echo</i>	29
<i>imagen 5: Nest Learning Thermostat</i>	29
<i>imagen 6: Elgato Eve Degree</i>	30
<i>imagen 7: Nest Cam IQ</i>	31
<i>imagen 8: Phillips Hue Ambiance</i>	31
<i>imagen 9: Google Home</i>	32
<i>imagen 10: Elgato Eve Energy</i>	32
<i>imagen 11: Furbo</i>	33
<i>imagen 12: Nest Protect</i>	33
<i>imagen 13: Amazon Echo Dot</i>	34
<i>imagen 14: Netgear Arlo Pro</i>	35
<i>imagen 15: LG Smart Security Wireless Camera</i>	36
<i>imagen 16: August Smart Lock Homekit</i>	36
<i>imagen 17: Vivint Smart Home</i>	37
<i>imagen 18: SimpliSafe Home Security System</i>	37
<i>imagen 19: Ring Video Doorbell Pro</i>	38
<i>imagen 20: SkyBell HD WI-FI Video Doorbell</i>	39
<i>imagen 22: Sistemas operativos móviles en 2017</i>	39
<i>imagen 23: esquema de la interfaz</i>	49
<i>imagen 24: diagrama uml de CU_01</i>	49
<i>imagen 25: diagrama uml de CU_02</i>	50
<i>imagen 26: diagrama uml de CU_03</i>	50
<i>imagen 27: diagrama uml de CU_04</i>	51
<i>imagen 28: diagrama uml de CU_05</i>	51
<i>imagen 29: diagrama uml de CU_06</i>	52
<i>imagen 30: diagrama uml de CU_07</i>	52
<i>imagen 31: diagrama uml de CU_08</i>	53
<i>imagen 32: diagrama uml de CU_09</i>	53
<i>imagen 33: Diagrama de componentes</i>	62
<i>imagen 34: Pantalla de registro</i>	66
<i>imagen 35: Pantalla principal</i>	67
<i>imagen 36: Pantalla de control de la puerta</i>	68
<i>imagen 37: Pantalla de control de luces</i>	69
<i>imagen 38: Pantalla de control de la cámara</i>	70

<i>imagen 39: Diagrama de navegación.....</i>	<i>71</i>
<i>imagen 40: esquema diseño hardware</i>	<i>72</i>
<i>imagen 41: Distribución de pines de la Raspberry pi 3</i>	<i>73</i>
<i>imagen 42: componentes que intervienen en el sistema</i>	<i>78</i>
<i>imagen 43: Diagrama de Gantt [1/2]</i>	<i>91</i>
<i>imagen 44: Diagrama de Gantt [2/2]</i>	<i>92</i>
<i>imagen 45: electronic lock example</i>	<i>100</i>

Chapter 1: Introduction and objectives

In this chapter, we will discuss the motivation and objectives that have led to the development of this end-of-grade Project. We will start with the first phases of the design and planification until the end of projects.

1.1 General Vision

The objective of this Project is to develop a home automation system controlled by an Android mobile phone application via WI-FI, specific, the home automation system will be restricted to the área around the door of the user's house.

The idea of the Project is to give the user the ability to watch who is at his door from his cell phone from the images generated from a camera plugged to the home automation system, and also giving him the ability to open and close his door as he wishes to, without the need to be physically there.



imagen 1: general idea of the system.

The system also includes an automatic mode for the home automation system, to help the user when it's dark outside, by turning on a light when a movement sensor activates, and the light sensor decides it's dark outside.

The project is divided in three different parts, that when they are put together, they form the complete system. The parts are a mobile phone application for the user to control all the system, the home automation system that controls the hardware, and the communication protocol that communicates these two other parts.

1.2 Motivation

We are currently in an era in which technological development is booming, and there are many services that have already been modernized to help and offer greater possibilities to users.

One of the most noticeable environments for increased technology has been in users' own homes, from programmable washers to smart TVs, our homes are gradually adding more and more devices that include their own integrated computers to Streamline their tasks.

However, the device that has undergone the most update in these years, has been without a doubt the mobile device, which has become almost indispensable in our lives.

Therefore, in this project we try to solve some of the problems of the users taking advantage of this technological development.

The main problem that we try to solve, or alleviate, is the problem with the keys to be able to open the lock of the homes, it is quite common, that, for some misunderstanding, you leave the keys of your own house in some other place, or that You're going to have a visitor, and you're not going to be able to wait to open the door for them.

With this project, we intend to give the user the possibility of controlling the door of his home from his mobile device, which can observe who is at the door and open it if he deems it necessary.

1.3 Objectives

The objective of the project is to implement a system that allows the user to control the environment of the door of his house from his mobile device. This general objective can be divided into smaller objectives, which, if completed, will allow the development of a prototype for the system.

- Development of a mobile application to allow the user to control the system.
- Development of a real-time system to control the home automation system, ensuring execution times.
- Ensure communication between the mobile application and the home automation system.

We also have some sub-objectives we would like to achieve while securing the achievement of the main objectives:

- We would like to make a user-friendly Android application which doesn't require the user to make extra efforts.

- The user's personal data has to be stored correctly encrypted.
- A real-time system has to be developed in order to control the home system.
- High error tolerance has to be achieved throughout the system.

1.4 Development phases

This section intends to expose the phases in which the development of the project has been divided:

The first phase consists in evaluating the different existing platforms to contain our system, in order to carry out the study of the feasibility of the system. Based on this, we choose the platform and architecture that best fits our system.

Once we know that the system is viable, we proceed to generate an analysis of the requirements that will define the behavior of the system. The analysis determines which requirements are possible and which are not, besides generating new requirements evaluating the main problem of the door.

With the requirements defined, you can begin to design the architecture, distributing the different parts of it among the elements that will compose our system, so that everything is well coupled.

At this point, we proceed to implement the system, both the mobile application and the home automation system. The elements are implemented separately, and finally they come together.

Once the entire system is implemented, we proceed to design and test the whole system, to ensure that there is no wrong or uncontrolled behavior.

The last part consists of writing the current document.

1.5 Document's structure

This section provides a brief description of the different chapters that make up this document:

Chapter 1.

It provides the most general vision of the project, in this chapter the objectives and motivations that have given rise to this project are exposed. The aim is to provide the reader with a general idea of what is to be found in this document

Chapter 2.

It defines the technological environment in which the project is to be developed, the options contemplated, the selection criteria, and the final hardware and software platforms on which this project will be carried out.

Chapter 3.

This chapter explains requirements analysis, architecture analysis, design and implementation of the project, begins by explaining the design decisions taken, the requirements that have defined it and the tests performed on it

Chapter 4.

This chapter describes in-depth the implementation of the project have been made. We describe how we made the code for the Android application, and how we made the home smart system code, also including how the real-time system in charge of controlling the home automation system has been performed and of ensuring that the response times of the system comply with the requirements and requirements defined in the previous chapters of this document.

Chapter 5.

This chapter defines the planning and development process used for this project, as well as a deployment of all the costs and costs associated with it, also known as the project budget.

Chapter 6.

To finalize the document, the conclusions reached during the development of the document, as well as the future lines of research and development that are intended to follow once this project is finally completed, are presented.

Capítulo 2: Entorno tecnológico

En este capítulo, se evaluará todo lo relacionado con la tecnología que incluye el proyecto. Se analizará la evolución que ha sufrido dicha tecnología y se compararan las diferentes opciones que existen para soportar el proyecto y ser capaces de crear el mejor producto posible.

2.1 Evolución de la tecnología

El comienzo de la domótica fue hace más de 30 años, en la época de los 80, que fue cuando las redes informáticas empezaron a desarrollar los sistemas conocidos como redes informáticas de cableado, que más tarde se mejoraron con el uso de Wi-Fi.

Dicho avance tecnológico, sirvió para suplir las carencias existentes en el hogar que se daban en los años 80, ya que estos avances permitían integrar de manera eficiente todos los dispositivos electrónicos que se tuvieran en las casas, de esta manera, comenzaron a llegar a los hogares.

A principios de los noventa, comienza la era de la TIC (tecnología de informática y comunicaciones), que mejora considerablemente las posibilidades de la instalación domótica.

En la actualidad, con los sistemas de desarrollo 2.0, permiten formar sistemas domótico que requieren una menor tasa de envío de datos que los sistemas anteriores, posibilitando el mayor uso de la domótica, aumentando la seguridad de las comunicaciones y la velocidad de las mismas



imagen 2: ejemplo de sistema domótico

En el concepto de la tecnología de comunicaciones, existen, como en la mayoría de los desarrollos tecnológicos, una parte física, y otra lógica, ambas necesarias ya que una no tiene sentido de la otra, y sus desarrollos van normalmente de la mano.

El mercado de los dispositivos móviles ha sido uno de los que ha sufrido un crecimiento más rápido en las últimas décadas, ya que, en la actualidad hay una mayor cantidad de gente que tiene acceso a un dispositivo móvil de las que tienen acceso a un retrete.

Aunque el término “dispositivo móvil” abarca una amplitud de dispositivos diferentes, con características similares, para este proyecto nos interesan únicamente los smartphones, aunque en un futuro, sería posible aumentar el mercado a el resto de dispositivos existentes.

Desde el año 2002, el uso de la telefonía móvil se ha disparado, sobre todo desde la llegada de los teléfonos inteligentes, o smartphones, siendo así que, en 2014, la navegación en internet desde dispositivos móviles supero a la de ordenadores de sobremesa y portátiles.

Aunque, con el aumento de usuarios de estos dispositivos, el uso que se les da ha cambiado radicalmente. Actualmente el mayor uso que se da a estos dispositivos es la mensajería instantánea y el uso de redes sociales.

Es este desarrollo tecnológico el que ha permitido que actualmente podamos hacer prácticamente cualquier cosa desde nuestro dispositivo móvil, y es lo que ha hecho posible este proyecto.

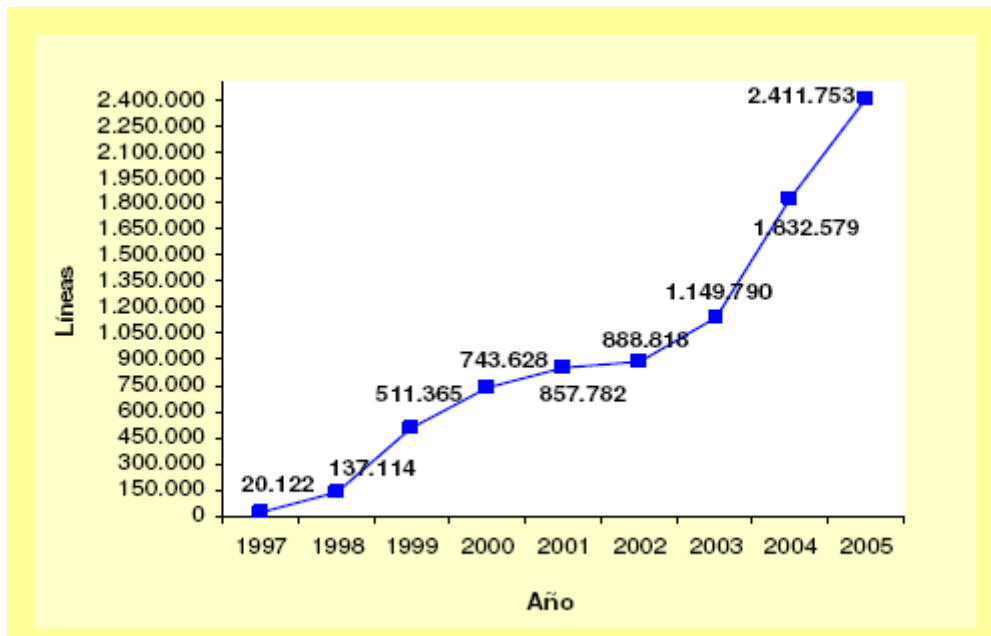


imagen 3: uso histórico de la telefonía móvil

2.2 sistemas domóticos más utilizados actualmente.

Como hemos mencionado previamente, los sistemas domóticos están en auge ahora mismo, por lo que en esta sección vamos a exponer los 10 sistemas domóticos más utilizados en la actualidad, para que sirvan como ejemplo.

1. Amazon Echo:

Este producto desarrollado por Amazon es un altavoz controlado por voz con su propio asistente virtual, llamado Alexa. Basta con decir, "Alexa, aumente la calefacción en la sala de estar" y si se cuenta con el termostato adecuado Alexa subirá la temperatura. Ella también le dirá una broma, confirmar los resultados deportivos y así sucesivamente. Los altavoces multidireccionales del Echo también se pueden utilizar para reproducir música, aunque no es de alta calidad. En el caso de tener una persona llamada Alexa en su hogar, puede cambiar la "palabra de alerta" que hace que el sistema le escuche.



imagen 4: Amazon Echo

2. Nest Learning Thermostat:

Nest, otro producto de Amazon es un controlador inalámbrico para los radiadores y calefactores de la casa que puede ser montado en la pared o colocado en un soporte portátil. Si el sensor se da cuenta de que ha salido de la casa cambia al modo Ausente, para ahorrar energía, aunque también puede activar o desactivar la calefacción y el agua remotamente a través del teléfono inteligente. El sistema le anima suavemente a bajar la temperatura y le envía un resumen de energía cada mes. Este producto te ofrece la posibilidad de fijar un horario, aunque en los primeros días se nota cuando se enciende y apaga la calefacción.



imagen 5: Nest Learning Thermostat

3. Elgato Eve Degree:

Este producto de Eve es un sensor de meteorología atractivo y minúsculo. Tiene una caja de aluminio anodizado y resulta fácil de configurar: se enciende, descargas la aplicación Elgato Eve y apuntas la cámara de tu iPhone (solo es compatible con HomeKit) a el código impreso en la parte inferior, y ya está listo para funcionar. El grado tiene resistencia al agua por lo que se puede colocar al aire libre o en. Mide la humedad, la temperatura y la presión atmosférica. Se puede utilizar de forma remota utilizando la excelente aplicación de Eve o Siri, aplicación que viene incluida en todos los dispositivos Apple. Simplemente hay que decirle a tu iPhone, "Hey Siri, ¿cuál es la temperatura en la sala de estar?" y este le responderá el valor.



imagen 6: Elgato Eve Degree

4. Nest Cam IQ:

El IQ es la última cámara de seguridad de Nest. Cuenta con una lente de alta calidad y un sensor digital lo que implica que, por ejemplo, si ve a un intruso, puede alertarlo y transmitir imágenes a una resolución lo suficientemente alta como para que la evidencia sea útil. El software inteligente también significa que el vídeo transmitido puede acercarse a una persona y seguirla por la habitación. Tiene instalado una aplicación de reconocimiento facial, disponible como parte de Nest Aware, pero que requiere una suscripción anual. Puede ver en la oscuridad y cuenta con un altavoz y un micrófono, lo que significa que puede comunicarse con el hogar desde su teléfono.



imagen 7: Nest Cam IQ

5. Phillips Hue Ambiance:

Hue es uno de los últimos productos de la marca Phillips. Este producto cuenta con tres bombillas en que se controlan desde una unidad puente inalámbrica. Los bulbos cuentan con una variedad de 16.000.000 de colores. Cuenta con la posibilidad de encender y apagar las luces de forma remota, y además agregar otras luces de la a el sistema.



imagen 8: Phillips Hue Ambiance

6. Google Home:

este producto es el altavoz inteligente que ha desarrollado Google. Cuenta con una función muy parecida al Echo de Amazon, pero la frase de activación en este caso es “ok, Google”. El reconocimiento de voz es bastante preciso. Puede reproducir música, establecer alarmas, crear una lista de la compra o decirte el tiempo que te va a llevar llegar a tu destino. Puede interactuar con otros productos de la gama de SmartThings de Samsung.



imagen 9: Google Home

7. Elgato Eve Energy:

Este producto de Eve es un interruptor de alimentación inteligente, que es una de las formas más sencillas de entrar en la automatización del hogar. Este dispositivo funciona con los productos HomeKit, se conecta a su toma de corriente y luego conecta cualquier producto de esta gama que este enchufado a la corriente, y te permite interactuar con ellos desde tu teléfono móvil.



imagen 10: Elgato Eve Energy

8. Furbo:

Furbo es una cámara inteligente que permite ver lo que tu perro está haciendo, siempre que este delante de la cámara. Cuenta con un micrófono y un altavoz que te permiten hablar con el perro y obtener una alerta cuando el perro ladra. Furbo tiene un truco más en la manga: antes de salir, puedes llenar el dispositivo con golosinas. Entonces, con pulsar un botón de la aplicación de móvil, Furbo expulsará un regalo para su perro.



imagen 11: Furbo

9. Nest Protect:

Esta es una alarma de humo y monóxido de carbono que cuenta con un sensor inteligente, que puede distinguir la diferencia entre el humo y el vapor. En lugar del ruido estridente y desagradable que hacen la mayoría de las alarmas de humo, éste emite una advertencia hablada antes de que la alarma se apague. También emite una alerta para tu teléfono. Viene en dos versiones, con cable y con pilas. Si tiene la versión de la batería, te avisa cuando necesita reemplazar la batería. Si tiene el termostato Nest, ambos pueden trabajar juntos, así que, si Protect detecta una emergencia, el termostato apaga la calefacción. Incluso funciona como una luz por la noche.



imagen 12: Nest Protect

10. Amazon Echo Dot:

Este producto es un dispositivo que actúa como puente entre su dispositivo móvil y su reproductor de música. No está diseñado para ser un altavoz de música como el eco, pero en su lugar se puede conectar a altavoces de alta fidelidad para que pueda reproducir música a través de ellos en su lugar. Al igual que el eco más grande, puede pedirle que lea las noticias, el horario de hoy, establecer un temporizador mientras

está cocinando o como una alarma en el dormitorio. El audio es lo suficientemente bueno para reproducir la radio, por ejemplo.



imagen 13: Amazon Echo Dot

2.3 sistemas domóticos más utilizados en el ámbito de seguridad.

Ya hemos mostrado cuales son los sistemas domoticos que tienen mayor éxito en la actualidad, pero solo uno de los 10 productos que hemos mostrado anteriormente se podría considerar que entra en la definición de “aplicaciones de seguridad”. Dado que este proyecto está considerado dentro de este ámbito, a continuación, mostraremos las aplicaciones domóticas más utilizadas en el ámbito de seguridad, descartando la “Nest Cam IQ” que estaría en el primer puesto, pero que ya la hemos descrito en el apartado anterior.

1. Netgear Arlo Pro

La cámara Netgear Arlo Pro, es una cámara de vigilancia que se conecta con su dispositivo móvil mediante una aplicación que han desarrollado. La cámara ofrece vídeo de 1.280 por 720 píxeles y tiene un campo de visión de 130 grados. Ofrece funcionalidad de zoom y utiliza ocho IR LEDs para proporcionar visión nocturna a 25 pies de distancia. También cuenta con un altavoz y micrófono para la comunicación bidireccional y la detección de sonido, así como un detector de movimiento. La parte superior de la cámara tiene dos botones; El más cercano al frente se utiliza para sincronizarlo con la estación base, y el que está hacia la parte posterior se utiliza para liberar el compartimento de la batería. Utiliza Wi-Fi 802.11n para comunicarse con la estación base y tiene una autonomía de hasta 300 pies. El Arlo Pro utiliza una batería recargable de iones de litio diseñada para durar hasta seis meses antes de requerir una recarga.



imagen 14: Netgear Arlo Pro

2. LG Smart Security Wireless Camera LH5200WI

Las cámaras de seguridad para el hogar vienen en todas las formas y tamaños, pero el LHC5200WI es fácilmente uno de los modelos más únicos que existe. La cámara en forma de cono negro tiene 3 pulgadas de alto y mide 5 pulgadas en la parte superior y 2 pulgadas en la base, donde encontrará un puerto de alimentación USB micro incorporado. Está diseñada para ser colocada en la intemperie, y tendrías que mirar muy de cerca para identificarlo como un dispositivo de seguridad.

La lente se asienta detrás de una cubierta negra cerca de la parte superior de la carcasa y está emparejada por dos LEDs IR que proporcionan hasta 15 pies de visión nocturna. Justo debajo de la lente hay una tira de LED delgada y curvada que parpadea en azul cuando está en modo Bluetooth, azul y verde cuando se conecta a Wi-Fi, verde fijo cuando una conexión Wi-Fi es exitosa, ámbar al descargar una actualización y ámbar y Verde cuando se instala una actualización.

Dentro de la carcasa hay un sensor de 5 megapíxeles con un campo de visión de 130 grados, radios Bluetooth y Wi-Fi, sensores de movimiento, temperatura y humedad, micrófono y altavoz para comunicación bidireccional y una sirena de 93dB. También hay circuitos Z-Wave para controlar dispositivos domésticos inteligentes como reguladores de luz, sensores de agua e interruptores inteligentes de Aetoe, Eaton y GE. A partir de esta escritura la lista de dispositivos compatibles se limitó a alrededor de diez, pero un portavoz prometió que más se agregará con el tiempo.



imagen 15: LG Smart Security Wireless Camera

3. August Smart Lock Homekit Enabled

Como el nombre de este sistema indica, este producto esta pensado para funcionar con la gama de productos Apple HomeKit, y puede ser controlado mediante el asistente de Apple "Siri". El sistema funciona a modo de pomo para la puerta, y sigue funcionando del modo normal, de manera que puedes abrir y cerrar la puerta simplemente girando el dispositivo. Además, incluye un sistema de cierre automático que bloquea la puerta una vez se ha abierto y ha pasado cierto tiempo, también puedes activar el sistema de geo cierre, que utiliza el gps de tu dispositivo móvil para abrir la puerta en base a tu posición. La aplicación móvil te permite recibir un correo electrónico cada vez que la puerta es utilizada.



imagen 16: August Smart Lock Homekit

4. Vivint Smart Home

Este dispositivo, no es un dispositivo de seguridad por si mismo, ya que únicamente funciona a modo de controlador general. La idea de este producto es construir un sistema de seguridad basado en las necesidades de cada usuario, para ello necesitas comprar una serie de sensores y actuadores para controlar los accesos a tu casa, y luego necesitas adquirir este producto para conectar y controlar todo el sistema.

Este sistema incluye además una aplicación para los dispositivos móviles desde la que puedes acceder a cualquier dispositivo que hayas conectado al sistema, de manera que, si tienes conectada, desde tu teléfono móvil podrás ver las imágenes que esta recoge.



imagen 17: Vivint Smart Home

5. SimpliSafe Home Security System

Este producto es muy similar al anterior, el producto base que venden es una estación base con un par de sensores conectados a él, pero, para tener un sistema de seguridad realmente útil, hay que adquirir más sensores de los que vienen incluidos en el paquete básico.

El paquete base, incluye la estación base, una pantalla inalámbrica, un sensor de movimiento y un sensor para la puerta. La estación base es la misma para todos los sensores, de manera que, si quisieras instalar adicionalmente una cámara de seguridad, podrías hacerlo fácilmente. Una de las pegas que tiene este sistema es que no cuenta con una aplicación móvil para controlar lo que sucede en la casa, por lo que no podrás conocer lo que sucede en ella a distancia.



imagen 18: SimpliSafe Home Security System

6. Ring Video Doorbell Pro

La definición más acertada para este producto sería un timbre inteligente. Este timbre viene instalado con una cámara y una serie de sensores, de manera que actúa de las siguientes maneras. Funciona como un timbre normal, es decir, tocan el botón y suena una alarma en la casa. La cámara emite video en alta definición en directo, por lo que puedes visualizar desde una pantalla situada en el interior de la vivienda quien está en la puerta. Como cuenta con conexión WI-FI, el sistema puede almacenar las imágenes que toma en la nube, si contratas este servicio adicional, pero, además, la conexión WI-FI permite emitir las imágenes a la aplicación móvil que puedes tener instalada en el móvil, de manera que veras en tiempo real desde tu teléfono quien está en la puerta.



imagen 19: Ring Video Doorbell Pro

7. SkyBell HD WI-FI Video Doorbell

Este producto, es, al igual que el anterior, un timbre inteligente. Cuenta con muchas de las características del anterior producto, video en alta definición, conexión con la aplicación móvil, y las funciones normales de un timbre. No cuenta con el servicio para almacenar las imágenes en la nube.

Pero, este producto cuenta además con un sensor de movimiento, que emite alertas al teléfono móvil cada vez que se active, por lo que podrás conocer y ver si hay alguien rondando tu puerta. Cuenta también con una serie de LEDs que permiten tomar imágenes en la oscuridad.



imagen 20: SkyBell HD WI-Fi Video Doorbell

2.4 Análisis de la plataforma móvil

Si analizamos el mercado actual de telefonía móvil, podemos observar que, aunque existen multitud de sistemas operativos para estos dispositivos, actualmente solo hay cuatro que sean mundialmente conocidos por los usuarios normales, y son: Android, IOS, Windows Phone y BlackBerry. Si analizamos estos 4 sistemas operativos observamos que entre Android y IOS acaparan un 98,4% del mercado, dejando al resto de sistemas operativos con un mero 1,6%. Por ello, únicamente analizaremos en profundidad estos dos sistemas operativos.

Operating System	4Q16 Units	4Q16 Market Share (%)	4Q15 Units	4Q15 Market Share (%)
Android	352,669.9	81.7	325,394.4	80.7
iOS	77,038.9	17.9	71,525.9	17.7
Windows	1,092.2	0.3	4,395.0	1.1
BlackBerry	207.9	0.0	906.9	0.2
Other OS	530.4	0.1	887.3	0.2
Total	431,539.3	100.0	403,109.4	100.0

Source: Gartner (February 2017)

imagen 21: Sistemas operativos móviles en 2017

- **ANDROID:** es el sistema operativo desarrollado por Google, tiene su núcleo basado en Linux y ocupa un 80,7% del mercado actual. Dispone de una gran

cantidad de desarrolladores de aplicaciones, su sistema operativo es libre y cuenta con una política para la aceptación de aplicaciones de terceros.

- **IOS:** es el sistema operativo desarrollado por Apple y ocupa un 17,7% del mercado actual. Fue lanzado para el iPhone en 2007 y desde entonces ha crecido en gran manera. Su sistema operativo únicamente puede ser instalado en el hardware creado por su propia compañía, y únicamente acepta aplicaciones realizadas por sus propios desarrolladores.

Criterios de elección

La elección del sistema operativo que vamos a utilizar vendrá determinado por los siguientes criterios:

- **Punto primero:** queremos que el sistema pueda llegar al mayor número de usuarios posible, ya que cualquier persona puede potencialmente actualizar la puerta de su hogar para instalar una cerradura electrónica controlada por un sistema domótico.
- **Punto segundo:** el proyecto está pensando para que siga creciendo con el tiempo, por lo que nos interesa un sistema que evolucione y permita cambios en las aplicaciones.
- **Punto tercero:** la aplicación pretende poder utilizarse en una gran cantidad de dispositivos diferentes en el futuro, por lo que nos interesa un sistema operativo con la mayor variedad de dispositivos diferentes disponibles.

En base a estos criterios, el sistema operativo elegido ha resultado ser Android, ya que cuenta con una cantidad de cliente casi 5 veces mayor, existen una mayor variedad de dispositivos que utilizan Android como sistema operativo en lugar de IOS, y este sistema va a continuar creciendo, permitiendo que nuestro sistema también lo haga.

2.5 Análisis de hardware

2.5.1 Análisis de la placa para soportar el sistema domótico.

El principal análisis que tenemos que realizar para elegir el hardware que va a formar parte de nuestro sistema, consiste en analizar el dispositivo que vamos a utilizar como soporte para nuestro sistema domótico.

Existen multitud de opciones que podrían servir para soportar nuestro sistema, pero nosotros únicamente vamos a analizar las dos opciones más comunes, y que son de uso general, de manera que una vez demos por finalizado completamente este producto, lo podamos utilizar otra vez.

Estas dos opciones son Raspberry pi 3, y Arduino uno:

En la siguiente tabla mostramos las principales características de las dos placas:

	Arduino Uno	Raspberry Pi 3
Precio	30\$	35\$
Tamaño	2.95"x2.10"	3.37"x2.125"
Procesador	ATMega 328	ARM11
Velocidad de reloj	16MHz	700MHz
Memoria RAM	2KB	256MB
Memoria Flash	32KB	(tarjeta SD)
Voltaje de entrada	7-12V	5V
Potencia minima	42mA (.3W)	700mA (3.5W)
Pines digitales	14	24
PWM	6	N/A
IDE desarrollo	Arduino tool	Linux
Ethernet	N/A	10/100
USB	N/A	2 USB 2.0
Salida de video	N/A	HDMI, compuesto
Salida de audio	N/A	HDMI, analógico

Tabla 1: diferencia entre Arduino y Raspberry

Como podemos observar en la tabla, la Raspberry es bastante más potente que el Arduino, mientras que de precio y tamaño son muy parecidos.

Dado que los factores que nos interesan son los siguientes:

- Lo más barato posible.
- Cuentos más pines mejor.
- Conexión a internet.
- La mayor velocidad de reloj posible.
- Varias opciones de salida.

En vista a estos factores, la diferencia de precio es despreciable, y la cantidad de pines también, pero en el resto de factores, Raspberry es bastante superior al Arduino, por lo que elegiremos la Raspberry para nuestro proyecto.

2.5.2 Sensores:

- En lo relativo a los sensores disponibles para la Raspberry Pi, dado que en un principio queremos desarrollar una maqueta del proyecto, no nos interesan los sensores de alta gama, que evidentemente ofrecen mejores resultados que los de gama baja, pero, para demostrar que el sistema funciona nos basta con estos últimos, y dado que nuestro presupuesto es limitado, elegiremos sensores de gama baja.

Los sensores que hemos elegido han sido los siguientes:

- **Sensor de movimiento hc-sr501**, es el que ofrece una mayor sensibilidad de los que actualmente se ofertan para la Raspberry Pi.
- **Sensor de luminosidad LDR**, este es un producto genérico utilizado normalmente para realizar prototipos.

2.5.3 Actuadores:

Para este proyecto vamos a utilizar dos tipos de actuadores:

- **LED**: son diodos que emiten luz, utilizaremos 2, uno para simular la luz exterior de la casa del usuario y otro para simular la puerta de la misma.
- **Cámara**: En el proyecto utilizamos un módulo cámara de 5 megapíxeles que está diseñado específicamente para Raspberry Pi, con un lente de foco fijo. Es capaz de tomar imágenes estáticas de 2592 x 1944, y también es compatible con el formato de video 1080p30, 720p60 y 640x480p60/90. Se conecta al Raspberry Pi por medio de un pequeño conector en la parte superior de la tarjeta y utiliza la interfaz dedicada CSI, diseñado especialmente para la conexión de cámaras.

2.6 Tecnología utilizada

- **Entorno de desarrollo de la aplicación móvil: Android Studio**
- **Lenguaje de desarrollo de la aplicación móvil: Java**
- **Android SDK**
- **Lenguaje de desarrollo del sistema domótico: C**

Capítulo 3: Análisis diseño del sistema

En este capítulo, se van a especificar las decisiones tomadas respecto al análisis del sistema, los requisitos de este y el diseño de la aplicación. Cada uno de las siguientes secciones detalla uno de estos elementos en concreto.

3.1: Análisis

3.1.1: Requisitos de usuario

En esta sección se encuentran reunidos todos los requisitos de usuario que han de ser implementados en el código, estos requisitos se dividen en requisitos de interfaz y de restricción.

Los requisitos contendrán los siguientes campos:

- **Identificador (ID):** este será el código identificativo para cada requisito, la denominación de este campo seguirá la siguiente nomenclatura: RU_XX_YY, donde RU indica que es un requisito de usuario, XX será CA en el caso de capacidad, y RE en el caso de restricción, finalmente YY se sustituirá por un número que se incrementará con cada nuevo requisito del mismo tipo, comenzando por el 01.
- **Necesidad:** indica la importancia que tiene el requisito para el sistema, puede ser alta, media o baja.
- **Estabilidad:** indica las posibilidades que tiene el requisito de ser cambiado durante el desarrollo del proyecto, puede tomar los valores estable e inestable.
- **Verificabilidad:** indica la facilidad con la que puede comprobarse que el requisito este correctamente implementado en el sistema. Sus valores posibles serán alta, media y baja.
- **Descripción:** describe de manera clara y concisa el requisito que se esté tratando.

Requisitos de capacidad:

ID	RU_CA_01	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	La aplicación móvil contara con una pantalla de inicio donde el usuario pueda registrarse.		

Tabla 2: RU_CA_01

ID	RU_CA_02	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	La pantalla de inicio contara con los siguientes campos para el registro: <i>usuario</i> y <i>contraseña</i> , y un botón <i>acceder</i> .		

Tabla 3: RU_CA_02

ID	RU_CA_03	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	En la pantalla inicial el botón de <i>acceder</i> deberá dirigir al usuario a la pantalla principal.		

Tabla 4: RU_CA_03

ID	RU_CA_04	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	La pantalla principal de la aplicación contara con 3 botones que serán los siguientes: <i>control de luz, cámara y control de la puerta</i> .		

Tabla 5: RU_CA_04

ID	RU_CA_05	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	En la pantalla principal el botón de <i>control de luz</i> deberá dirigir al usuario a la pantalla de luz.		

Tabla 6: RU_CA_05

ID	RU_CA_06	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	En la pantalla principal el botón de <i>cámara</i> deberá dirigir al usuario a la pantalla de la cámara.		

Tabla 7: RU_CA_06

ID	RU_CA_07	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	En la pantalla principal el botón de <i>control de la puerta</i> deberá dirigir al usuario a la pantalla para el control de la puerta.		

Tabla 8: RU_CA_07

ID	RU_CA_08	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	La pantalla de control de luz contara con los siguientes botones: <i>Encender luz, apagar luz, activar, desactivar y volver.</i>		

Tabla 9: RU_CA_08

ID	RU_CA_09	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	La pantalla de control de la puerta contara con los siguientes botones: <i>abrir puerta</i> , <i>cerrar puerta</i> y <i>volver</i>		

Tabla 10: RU_CA_09

ID	RU_CA_10	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Baja.
Descripción	La pantalla de la cámara, mostrara la última imagen tomada por la cámara y contara con un botón <i>otra foto</i> , y <i>volver</i>		

Tabla 11: RU_CA_10

ID	RU_CA_11	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	En la pantalla de control de luz, el botón de <i>encender luz</i> deberá encender la luz exterior, y el botón de <i>apagar luz</i> deberá apagarla.		

Tabla 12: RU_CA_11

ID	RU_CA_12	Necesidad	baja
Estabilidad	inestable	Verificabilidad	Alta
Descripción	En la pantalla de control de luz, el botón de <i>activar</i> deberá habilitar las luces.		

Tabla 13: RU_CA_12

ID	RU_CA_13	Necesidad	baja
Estabilidad	inestable	Verificabilidad	Alta
Descripción	En la pantalla de control de luz, el botón de <i>desactivar</i> deberá deshabilitar el estado de las luces.		

Tabla 14: RU_CA_13

ID	RU_CA_14	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	En la pantalla de control de la puerta, el botón de <i>abrir puerta</i> deberá abrir la puerta, y el botón de <i>cerrar puerta</i> deberá cerrarla.		

Tabla 15: RU_CA_14

ID	RU_CA_15	Necesidad	media
Estabilidad	estable	Verificabilidad	Alta
Descripción	Todos los botones denominados <i>volver</i> deberán dirigir al usuario a la anterior pantalla visitada.		

Tabla 16: RU_CA_15

ID	RU_CA_16	Necesidad	Alta
Estabilidad	estable	Verificabilidad	Alta
Descripción	En la pantalla de cámara, el botón de <i>otra foto</i> deberá generar una nueva imagen con la cámara.		

Tabla 17: RU_CA_16

Requisitos de restricción:

ID	RU_RE_01	Necesidad	media
Estabilidad	estable	Verificabilidad	Alta
Descripción	El usuario no podrá encender las luces si estas no están habilitadas.		

Tabla 18: RU_RE_01

ID	RU_RE_02	Necesidad	media
Estabilidad	estable	Verificabilidad	Alta
Descripción	En la pantalla del control de luz, el usuario no podrá pulsar <i>activar</i> a no ser que las luces hayan sido deshabilitadas previamente.		

Tabla 19: RU_RE_02

ID	RU_RE_03	Necesidad	media
Estabilidad	estable	Verificabilidad	Alta
Descripción	En la pantalla del control de luz, el usuario no podrá pulsar <i>desactivar</i> a no ser que las luces hayan sido habilitadas previamente.		

Tabla 20: RU_RE_03

3.1.2: Casos de uso

En este apartado, detallaremos los posibles caminos que puede tomar un usuario mientras utiliza nuestro sistema, para ello detallaremos los elementos necesarios para que ocurran estas situaciones.

Las tablas en las que vamos a plasmar estos casos de uso contarán con los siguientes campos:

- **ID:** representa el identificador del caso de uso, su nomenclatura será CU_XX, donde XX será un número de dos cifras que se incrementará con cada nuevo caso de uso, comenzando por el 01.
- **Objetivo:** describe la meta a la que se quiere llegar con este caso de uso.
- **Precondiciones:** condiciones que se tienen que dar para que el caso de uso sea válido.
- **Descripción:** pasos que realizara el usuario para alcanzar su meta.
- **Postcondiciones:** condiciones que se darán si se completa el caso de uso.

Además, se incluirá un diagrama de casos UML para cada uno de los casos de uso.

A continuación, mostramos un diagrama donde se representa la interfaz completa de nuestra aplicación móvil:

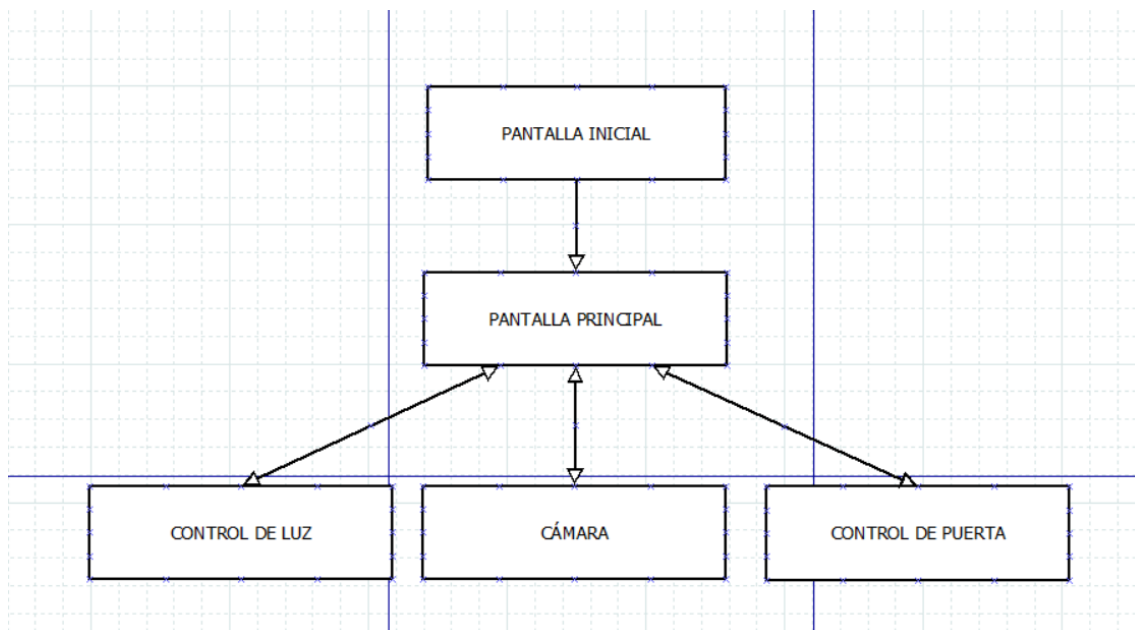


imagen 22: esquema de la interfaz

ID	CU_01	Objetivo	Entrar en la aplicación
Precondiciones	-conexión a internet -conocimiento de las credenciales	Postcondiciones	-se obtiene acceso a la aplicación.
Descripción	El usuario abre la aplicación, introduce el usuario y la contraseña, pulsa <i>entrar</i> y accede a la aplicación, a la pantalla principal.		

Tabla 21: CU_01

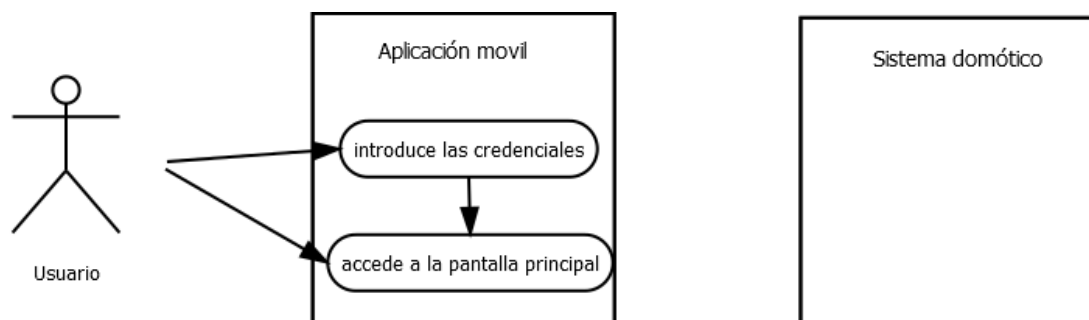


imagen 23: diagrama uml de CU_01

ID	CU_02	Objetivo	Encender la luz
Precondiciones	-conexión a internet -conocimiento de las credenciales -la luz no está deshabilitada	Postcondiciones	-la luz esta ahora encendida.
Descripción	Desde la pantalla principal, el usuario pulsa en <i>control de luz</i> , y una vez ha accedido a la pantalla, pulsa en <i>encender luz</i> .		

Tabla 22: CU_02

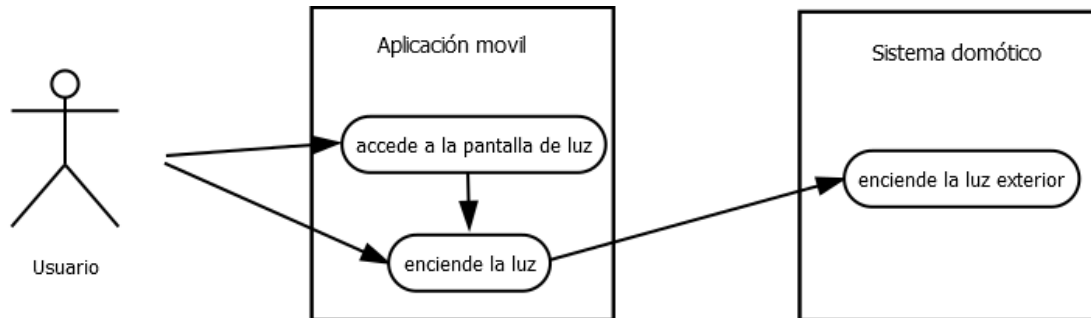


imagen 24: diagrama uml de CU_02

ID	CU_03	Objetivo	Apagar la luz
Precondiciones	-conexión a internet -conocimiento de las credenciales -la luz no está deshabilitada	Postcondiciones	-la luz esta ahora apagada.
Descripción	Desde la pantalla principal, el usuario pulsa en <i>control de luz</i> , y una vez ha accedido a la pantalla, pulsa en <i>apagar luz</i> .		

Tabla 23: CU_03

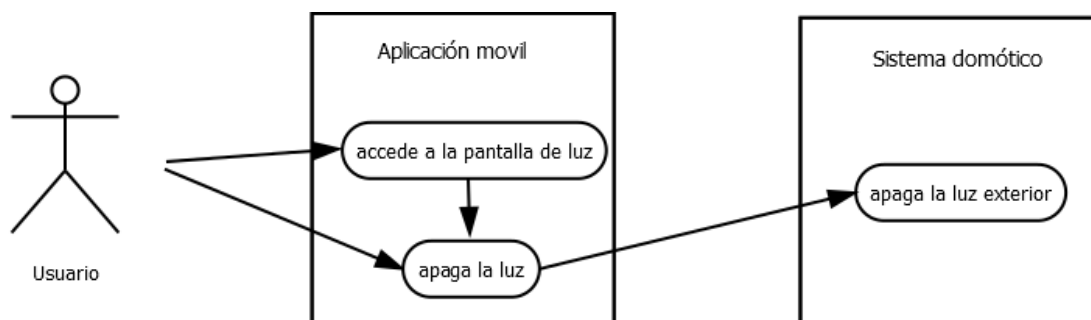


imagen 25: diagrama uml de CU_03

ID	CU_04	Objetivo	Mirar la cámara
Precondiciones	-conexión a internet -conocimiento de las credenciales	Postcondiciones	-no aplica
Descripción	El usuario pulsa sobre el botón <i>cámara</i> en la pantalla principal, una vez en la pantalla de la cámara, aparecerá la imagen.		

Tabla 24: CU_04

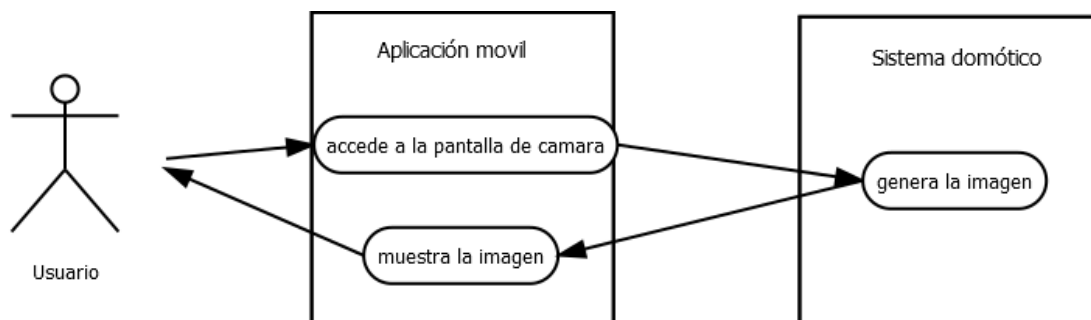


imagen 26: diagrama uml de CU_04

ID	CU_05	Objetivo	Abrir la puerta
Precondiciones	-conexión a internet -conocimiento de las credenciales	Postcondiciones	-la puerta se abre
Descripción	El usuario pulsa sobre el botón <i>control de la puerta</i> en la pantalla principal, una vez en la pantalla de la puerta, pulsa el botón <i>abrir puerta</i> .		

Tabla 25: CU_05

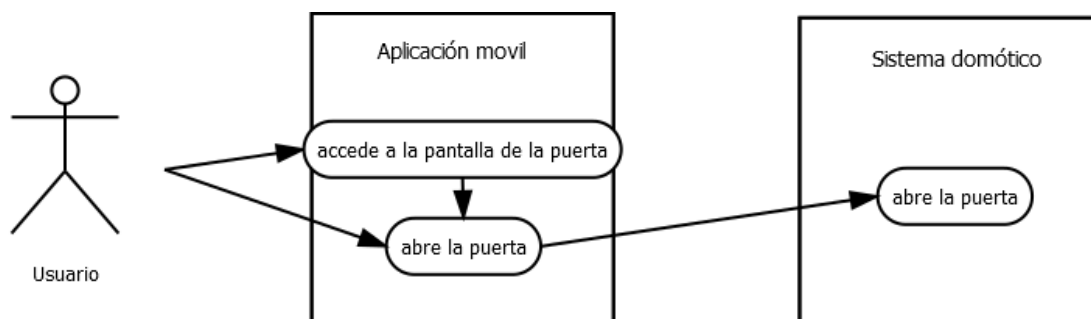


imagen 27: diagrama uml de CU_05

ID	CU_06	Objetivo	Cerrar la puerta
Precondiciones	-conexión a internet -conocimiento de las credenciales	Postcondiciones	-la puerta se cierra
Descripción	El usuario pulsa sobre el botón <i>control de la puerta</i> en la pantalla principal, una vez en la pantalla de la puerta, pulsa el botón <i>cerrar puerta</i> .		

Tabla 26: CU_06

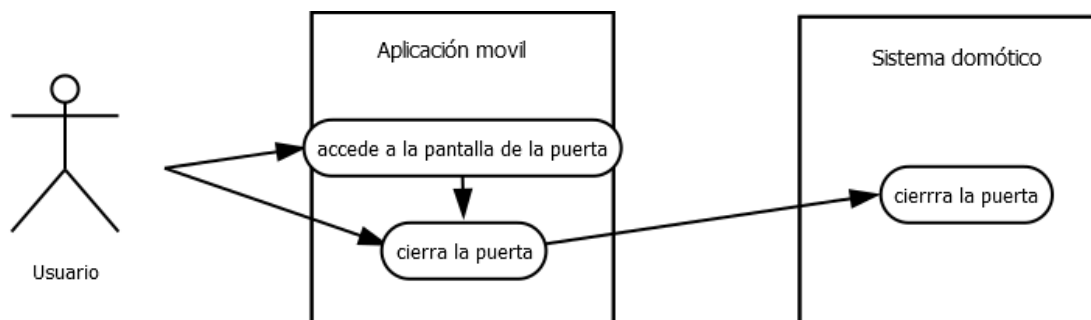


imagen 28: diagrama uml de CU_06

ID	CU_07	Objetivo	Deshabilitar las luces
Precondiciones	-conexión a internet -conocimiento de las credenciales -la luz no está deshabilitada	Postcondiciones	-la luz ya no podrá ser utilizada
Descripción	Desde la pantalla principal, el usuario pulsa en <i>control de luz</i> , y una vez ha accedido a la pantalla, pulsa en <i>deshabilitar luz</i> .		

Tabla 27: CU_07

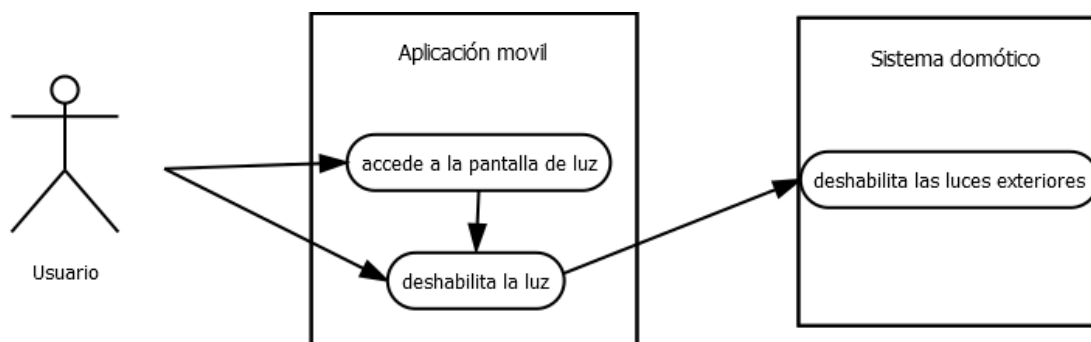


imagen 29: diagrama uml de CU_07

ID	CU_08	Objetivo	Habilitar las luces
Precondiciones	-conexión a internet -conocimiento de las credenciales -la luz no está habilitada	Postcondiciones	-la luz podrá ser utilizada.
Descripción	Desde la pantalla principal, el usuario pulsa en <i>control de luz</i> , y una vez ha accedido a la pantalla, pulsa en <i>habilitar luz</i> .		

Tabla 28: CU_08

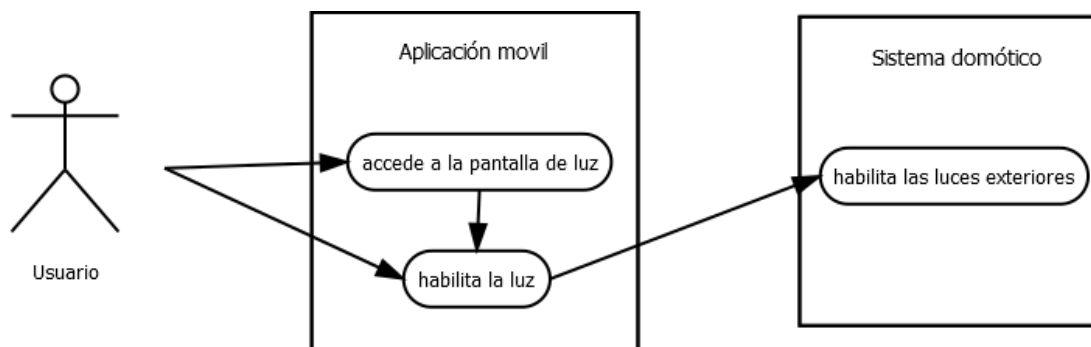


imagen 30: diagrama uml de CU_08

ID	CU_09	Objetivo	Generar otra imagen con la cámara
Precondiciones	-conexión a internet -conocimiento de las credenciales	Postcondiciones	-no aplica
Descripción	El usuario pulsa sobre el botón <i>cámara</i> en la pantalla principal, una vez en la pantalla de la cámara, pulsa en <i>otra foto</i> .		

Tabla 29: CU_09

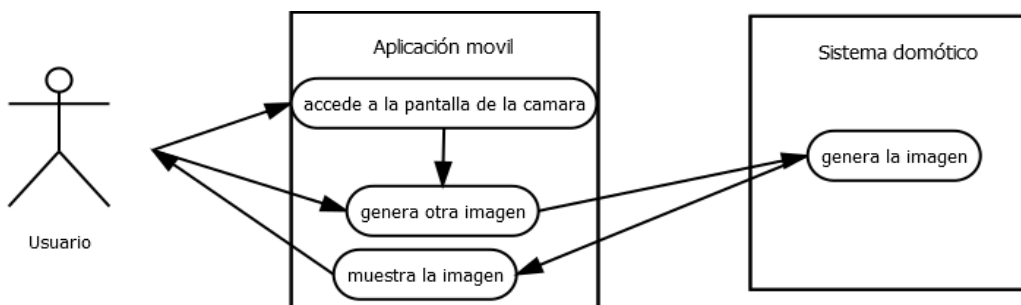


imagen 31: diagrama uml de CU_09

3.1.3: Requisitos software

En esta sección se encuentran reunidos todos los requisitos funcionales que detallan como deben ser los requisitos de usuario implementados en el código, se encontraran divididos en requisitos funcionales y no funcionales.

Los requisitos se van a recoger en tablas que seguirán la siguiente nomenclatura:

- **Identificador (ID):** este será el código identificativo para cada requisito, la denominación de este campo seguirá la siguiente nomenclatura: RS_XX_YY, donde RS indica que es un requisito de usuario, XX será FU en el caso de funcionales, y NF en el caso de no funcionales, finalmente YY se sustituirá por un número que se incrementará con cada nuevo requisito del mismo tipo, comenzando por el 01.
- **Necesidad:** indica la importancia que tiene el requisito para el sistema, puede ser alta, media o baja.
- **Estabilidad:** indica las posibilidades que tiene el requisito de ser cambiado durante el desarrollo del proyecto, puede tomar los valores estable e inestable.
- **Verificabilidad:** indica la facilidad con la que puede comprobarse que el requisito este correctamente implementado en el sistema. Sus valores posibles serán alta, media y baja.
- **Descripción:** describe de manera clara y concisa el requisito que se esté tratando.

Requisitos funcionales:

ID	RS_FU_01	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	Cuando se lanza la aplicación móvil, la primera pantalla debe ser una pantalla de registro.		

Tabla 30: RS_FU_01

ID	RS_FU_02	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	La pantalla de registro contara con dos campos de texto, uno para el usuario, y otro para la contraseña, además contara con un botón.		

Tabla 31: RS_FU_02

ID	RS_FU_03	Necesidad	alta
Estabilidad	estable	Verificabilidad	media
Descripción	Una vez pulsado el botón de la pantalla inicial, se deben comprobar si las credenciales son correctas. En caso de que lo sean, se debe redirigir al usuario a la pantalla principal.		

Tabla 32: RS_FU_03

ID	RS_FU_04	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	La pantalla principal contara con tres botones, uno para cada tipo de funcionalidad: cámara, luz y puerta.		

Tabla 33: RS_FU_04

ID	RS_FU_05	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	Una vez pulsado el botón para acceder a la puerta, se debe cambiar automáticamente a dicha pantalla.		

Tabla 34: RS_FU_05

ID	RS_FU_06	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	Una vez pulsado el botón para acceder a las luces, se debe cambiar automáticamente a dicha pantalla.		

Tabla 35: RS_FU_06

ID	RS_FU_07	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	Una vez pulsado el botón para acceder a la cámara, se debe cambiar automáticamente a dicha pantalla.		

Tabla 36: RS_FU_07

ID	RS_FU_08	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	La pantalla del control de luz contara con 5 botones, dos para apagar y encender las luces, dos para deshabilitar las luces, y otro para volver a la pantalla principal.		

Tabla 37: RS_FU_08

ID	RS_FU_09	Necesidad	baja
Estabilidad	inestable	Verificabilidad	alta
Descripción	Si las luces están encendidas, se debe marcar dicho botón.		

Tabla 38: RS_FU_09

ID	RS_FU_10	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	Los botones de encendido y apagado de luces mandaran una señal al sistema domótico para que apague la luz.		

Tabla 39: RS_FU_10

ID	RS_FU_11	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	Los botones para habilitar y deshabilitar las luces mandaran una señal al sistema domótico para que reanude o detenga la interacción con las luces.		

Tabla 40: RS_FU_11

ID	RS_FU_12	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	El estado de las luces cuando están deshabilitadas será apagado.		

Tabla 41: RS_FU_12

ID	RS_FU_13	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	La pantalla del control de la puerta contara con 3 botones, dos para abrir y cerrar la puerta y otro para volver a la pantalla principal.		

Tabla 42: RS_FU_13

ID	RS_FU_14	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	Los botones para abrir y cerrar la puerta enviaran una señal al sistema domótico para que realice la acción pertinente.		

Tabla 43: RS_FU_14

ID	RS_FU_15	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	La pantalla la cámara contara con 2 botones, uno para refrescar la imagen, y otro para volver a la pantalla principal, además contara con un campo para mostrar la imagen de la cámara.		

Tabla 44: RS_FU_15

ID	RS_FU_16	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	El botón de refrescar la cámara, mandara una señal al sistema domótico para que saque otra foto con la cámara, la envíe a la aplicación móvil, y esta lo muestre en su campo.		

Tabla 45: RS_FU_16

ID	RS_FU_17	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	Todos los botones identificados con <i>volver</i> cambiaran la pantalla de la aplicación por la pantalla principal.		

Tabla 46: RS_FU_17

ID	RS_FU_18	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	El sistema domótico estará controlado por un sistema de tiempo real.		

Tabla 47: RS_FU_18

ID	RS_FU_19	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	El sensor de proximidad debe comprobarse cada medio segundo.		

Tabla 48: RS_FU_19

ID	RS_FU_20	Necesidad	alta
Estabilidad	estable	Verificabilidad	alta
Descripción	El sensor de luz debe comprobarse cada medio segundo.		

Tabla 49: RS_FU_20

ID	RS_FU_21	Necesidad	alta
Estabilidad	estable	Verificabilidad	baja
Descripción	Si la aplicación se cierra, esta debe enviar una señal al sistema domótico.		

Tabla 50: RS_FU_21

ID	RS_FU_22	Necesidad	baja
Estabilidad	inestable	Verificabilidad	media
Descripción	Cuando el sensor de movimiento se active, y hayan pasado más de 5 segundos desde su última activación, tomara una foto, y la enviara al sistema domótico,		

Tabla 51: RS_FU_22

Requisitos no funcionales:

ID	RS_NF_01	Necesidad	alta
Estabilidad	estable	Verificabilidad	baja
Descripción	La comunicación entre la aplicación y el sistema domótico será por vía wifi.		

Tabla 52: RS_NF_01

ID	RS_NF_02	Necesidad	media
Estabilidad	estable	Verificabilidad	alta
Descripción	La información sensible almacenada en la aplicación será almacenada cifrada.		

Tabla 53: RS_NF_02

ID	RS_NF_03	Necesidad	media
Estabilidad	estable	Verificabilidad	baja
Descripción	Toda la información que genere la aplicación, será almacenada hasta que se cierre la aplicación.		

Tabla 54: RS_NF_03

3.1.4: Matriz de trazabilidad

La siguiente matriz de trazabilidad se utilizará para comprobar que todos los requisitos de usuario han sido recogidos e implementados por al menos un requisito de software, de manera que, si alguna fila quedara sin marcar, implicaría que ese requisito no ha sido incluido:

	RS_FU_01	RS_FU_02	RS_FU_03	RS_FU_04	RS_FU_05	RS_FU_06	RS_FU_07	RS_FU_08	RS_FU_09	RS_FU_10	RS_FU_11	RS_FU_12	RS_FU_13	RS_FU_14	RS_FU_15
RU_IN_01	X	X	X												
RU_IN_02		X													
RU_IN_03			X												
RU_IN_04				X											
RU_IN_05						X									
RU_IN_06					X										
RU_IN_07							X								
RU_IN_08								X							
RU_IN_09													X		
RU_IN_10															X
RU_IN_11									X	X					
RU_IN_12											X				
RU_IN_13												X			
RU_IN_14														X	
RU_IN_15															
RU_IN_16															

Tabla 55: Matriz de trazabilidad [1/2]

	RS_FU_16	RS_FU_17	RS_FU_18	RS_FU_19	RS_FU_20	RS_FU_21	RS_FU_22	RS_NF_01	RS_NF_02	RS_NF_03
RU_IN_01										
RU_IN_02										
RU_IN_03										
RU_IN_04										
RU_IN_05										
RU_IN_06										
RU_IN_07										
RU_IN_08										
RU_IN_09										
RU_IN_10										
RU_IN_11										
RU_IN_12										
RU_IN_13										
RU_IN_14										
RU_IN_15		X								
RU_IN_16	X									

Tabla 56: Matriz de trazabilidad [2/2]

3.2: Diseño

En esta sección, explicaremos los elementos que componen el desarrollo del software implicado en nuestro sistema. Se analizará la arquitectura del sistema y las interfaces de la aplicación móvil con la que interacciona el usuario.

3.2.1: Diseño de la arquitectura

Este sistema está basado en una arquitectura maestro- Esclavo, esta arquitectura consiste en una aplicación maestro se encarga de enviar ordenes, en forma de peticiones a la aplicación esclavo, que realizara lo que venga especificado en la orden recibida.

En nuestro caso concreto, contamos con dos aplicaciones, la aplicación móvil que controla el usuario y que actúa como maestro, y el sistema domótico integrado en la raspberry pi, que será el esclavo. El maestro indicará al esclavo con que parte del sistema domótico quiere interactuar, y este deberá responder en consecuencia.

Van a existir tres tipos de orden, ordenes de encendido, apagado de elementos, ordenes de desactivación de elementos, y ordenes de recuperación de información. En los tres casos, la comunicación siempre la iniciara la aplicación maestro.

Se utilizará un sistema maestro esclavo con 3 niveles, que son:

- **Nivel 1:** corresponde a la aplicación móvil del cliente. Es la encargada de emitir ordenes en forma de solicitudes al sistema domótico, y cuenta con una interfaz para mostrar los datos al usuario y facilitar el uso de la misma.
- **Nivel 2:** corresponde al sistema domótico, cuya tarea consiste en escuchar a la aplicación móvil, y realizar las acciones solicitadas por este, además, deberá ser capaz de actuar de manera autónoma según lo definido en los requisitos y deberá controlar los sensores
- **Nivel 3:** son los sensores conectados al sistema domótico y que proporcionan toda la información al mismo

Para facilitar la implementación del sistema, y facilitar posibles cambios futuros, se utilizan patrones de diseño que aportan al sistema flexibilidad, posibilidad de evolución y facilidad en el mantenimiento.

El patrón que hemos utilizado para este sistema es el Modelo Vista-Controlador, que se adapta bastante bien a la arquitectura elegida. Este patrón separa las partes de las que se compone el sistema en componentes distintos, de manera que facilita el trabajo.

El patrón MVC cuenta con estos 3 elementos:

- **Modelo:** es el conjunto de clases lógicas que representa la información a ser reflejada en el mundo real.
- **Vista:** es el elemento encargado de mostrar la información al usuario.
- **Controlador:** este elemento es el encargado de recibir y tratar las comunicaciones y eventos producidos por los otros elementos del patrón.

El flujo que sigue el patrón MVC para funcionar en nuestro sistema es el siguiente:

1. La aplicación móvil realiza una petición al sistema domótico
2. El controlador recibe esta petición y la trata.
3. El controlador modifica si es necesario el modelo establecido.

4. El controlador envía la petición tratada a la vista.
5. La vista modifica la interfaz según establezca la petición.

Una vez tenemos definido como funciona el patrón y la arquitectura elegidas, procederemos a exponer los diferentes módulos en los que se va a dividir el sistema, de forma que obtengamos un diagrama de componentes.

En las siguientes figuras representaremos el sistema domótico y la aplicación móvil, cada separada en los módulos que la componen.

La relación entre los diferentes módulos está representada con flechas, y la dirección de las mismas representa el flujo de datos.

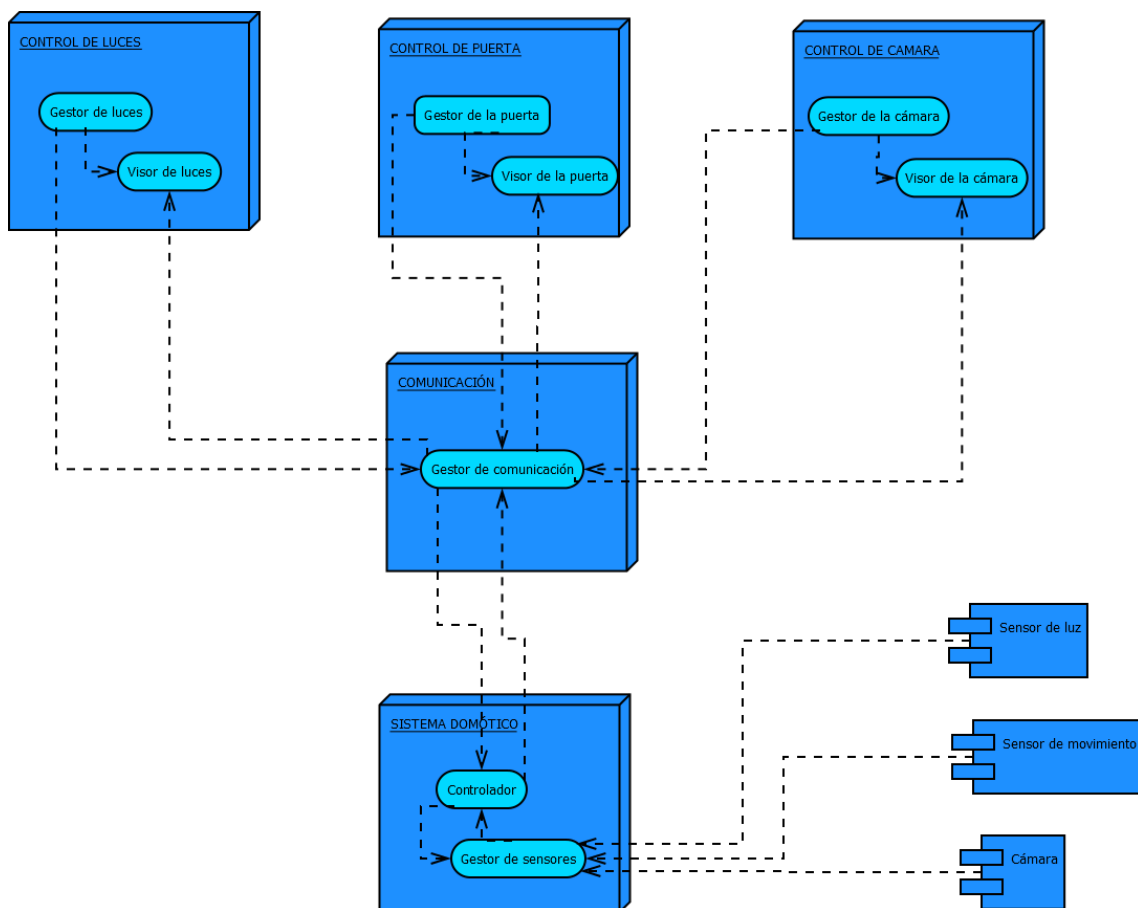


imagen 32:Diagrama de componentes

A continuación, vamos a detallar cada uno de los componentes con los que cuenta el sistema, las tablas cuentan con los siguientes campos:

- **Nombre:** nombre del componente
- **ID:** identificador único del componente, su nomenclatura sigue la siguiente pauta CO_XX, donde XX será sustituido por un numero incremental comenzando por 01.
- **Función:** función general que desempeña el componente
- **Tareas:** tareas asignadas a este componente
- **Referencias:** requisitos asociados con este componente

Gestor de luces	CO_01
Función	Controlar las luces
Tareas	<ul style="list-style-type: none"> • Encender luces • Apagar luces • Habilitar luces • Deshabilitar luces
Referencias	RS_FU_04, RS_FU_06, RS_FU_08, RS_FU_09, RS_FU_10, RS_FU_11, RS_FU_12

Tabla 57: CO_01 Gestor de luces

Visor de luces	CO_02
Función	Mostrar el estado de las luces
Tareas	<ul style="list-style-type: none"> • Mostrar luces encendidas • Mostrar luces apagadas • Mostrar luces habilitadas • Mostrar luces deshabilitadas
Referencias	RS_FU_04, RS_FU_06, RS_FU_08, RS_FU_09, RS_FU_10, RS_FU_11, RS_FU_12

Tabla 58: CO_02 Visor de luces

Gestor de puerta	CO_03
Función	Controlar la puerta
Tareas	<ul style="list-style-type: none"> • Abrir la puerta • Cerrar la puerta
Referencias	RS_FU_04, RS_FU_05, RS_FU_13, RS_FU_14

Tabla 59: CO_03 Gestor de puerta

Visor de puerta	CO_04
Función	Mostrar el estado de la puerta
Tareas	<ul style="list-style-type: none"> Mostrar puerta abierta Mostrar puerta cerrada
Referencias	RS_FU_04, RS_FU_05, RS_FU_13, RS_FU_14

Tabla 60: CO_04 Visor de puerta

Gestor de cámara	CO_05
Función	Controlar la cámara
Tareas	<ul style="list-style-type: none"> Tomar nueva imagen
Referencias	RS_FU_04, RS_FU_07, RS_FU_15, RS_FU_16

Tabla 61: CO_05 Gestor de cámara

Visor de cámara	CO_06
Función	Mostrar las imágenes de la cámara
Tareas	<ul style="list-style-type: none"> Mostrar imagen
Referencias	RS_FU_04, RS_FU_07, RS_FU_15, RS_FU_16

Tabla 62: CO_06 Visor de cámara

Gestor de comunicación	CO_07
Función	Comunicar la aplicación móvil con el sistema domótico
Tareas	<ul style="list-style-type: none"> Enviar instrucciones al sistema domótico: <ol style="list-style-type: none"> Control de luces Control de la puerta Tomar imagen Enviar datos a la aplicación móvil: <ol style="list-style-type: none"> Estado de luces Estado de la puerta Enviar imagen
Referencias	RS_FU_10, RS_FU_14, RS_FU_16, RS_FU_21, RS_NF_01

Tabla 63: CO_07 Gestor de comunicación

controlador	CO_08
Función	Controlar el sistema domótico
Tareas	<ul style="list-style-type: none"> • Recibir instrucciones • Controlar luces • Controlar puerta
Referencias	RS_FU_18, RS_FU_22

Tabla 64: CO_08 Controlador

Gestor de sensores	CO_09
Función	Obtener datos de los sensores
Tareas	<ul style="list-style-type: none"> • Leer los sensores • Tomar imagen • enviar datos
Referencias	RS_FU_19, RS_FU_20

Tabla 65: CO_09 Gestor de sensores

3.2.2: Diseño de la interfaz

En esta sección, explicaremos el proceso y las decisiones tomadas durante la creación de la interfaz para la aplicación móvil, ya que el sistema domótico no cuenta con ella. La interfaz ha sido diseñada pensando ante todo en el usuario final de la aplicación, de manera que la información aparezca de la forma más clara posible, que la aplicación sea intuitiva, es decir, que el usuario no tenga que aprender cómo funciona la aplicación, si no que pueda reconocer donde esta y como llegar a donde desea con facilidad, evitando todos los errores posibles durante el proceso.

A continuación, explicaremos pantalla por pantalla las decisiones de diseño tomadas:

3.2.2.1 Pantalla de registro



imagen 33: Pantalla de registro

Esta es una de las pantallas más importantes de la aplicación, ya que es la que establece quien puede acceder a ella, y quién no. Dado que el sistema que hemos diseñado, cuanta con la capacidad de abrir la puerta de la casa del usuario, o la puerta en la que se haya instalado el sistema, es muy importante que el usuario sepa que solo aquellos que él considere van a poder acceder a la puerta a través de nuestra aplicación.

A pesar de tener una gran importancia, no queremos que el usuario deje de utilizar la aplicación por culpa de la pantalla de registro, que podría pasar si esta requiere demasiado tiempo cada vez que el usuario accede a ella. Por lo que la interfaz es muy sencilla, únicamente tiene que introducir el usuario y la contraseña designados y pulsar el botón “entrar”.

Para el fondo de la aplicación, se ha utilizado una imagen con tonos grisáceos, de manera que no moleste al usuario para distinguir donde está la información que muestra la aplicación. Los botones de acción, es decir, los que interactúan con nuestro sistema domótico se han establecido con una tonalidad salmón, de manera que sean fácilmente reconocibles, así como los botones para la movilidad dentro de la página, se han establecido con una tonalidad azulada.

3.2.2.2 Pantalla principal



imagen 34: Pantalla principal

Esta es la pantalla principal de la aplicación, es la pantalla a la que accede el usuario una vez se ha registrado, y es la pantalla que conecta el resto de la aplicación.

La pantalla cuenta con 3 botones que se corresponden con las tres pantallas restantes de la aplicación: control de luces, control de cámara y control de la puerta. Siguiendo lo establecido en la pantalla de registro, dado que los botones son para navegar dentro de la aplicación cuentan con el tono azulado que los identifica.

Esta pantalla la añadimos para poder separar las funcionalidades de la aplicación, de manera que no sobrecargáramos al usuario con información nada más iniciar la misma, además, si se quisiera añadir más funcionalidad a la aplicación en un futuro, únicamente habría que añadir un botón extra para dicha funcionalidad en esta pantalla, sin necesidad de modificar las otras ya existentes.

3.2.2.3 Pantalla de control de la puerta



imagen 35: Pantalla de control de la puerta

Esta pantalla es la encargada de abrir y cerrar la puerta, cuenta con dos tipos de botones, los botones de acción para abrir y cerrar la puerta, en el tono salmon predefinido para ellos, y un boton “volver”, que devolvera al usuario a la pantalla principal, este botón al ser de navegacion por la aplicación, cuenta con el tono azulado.

Los botones de acción han sido situados en el centro de la pantalla de manera que la atencion del usuario se centre en ellos. El boton de volver esta situado en la esquina superior izquierda, de forma que el usuario pueda ver facilmente este boton, pero que no le reste atencion a los otros.

3.2.2.4 Pantalla de control de luces



imagen 36: Pantalla de control de luces

Esta pantalla es la encargada de controlar las luces, cuenta con dos tipos de botones, los botones de acción para encender, apagar, habilitar y deshabilitar las luces, en el

tono salmon predefinido para ellos, y un boton “volver”, que devolvera al usuario a la pantalla principal, este botón al ser de navegacion por la aplicación, cuenta con el tono azulado

3.2.2.5 Pantalla de control de la cámara



imagen 37: Pantalla de control de la cámara

Esta pantalla, conlleva una dificultad añadida con respecto a las dos anteriores, consiste en que se debe mostrar una imagen que ha generado la cámara conectada al sistema, de forma que el usuario pueda ver lo que sucede en el exterior de su vivienda.

La pantalla cuenta con dos botones, el botón volver que es idéntico al de las pantallas anteriores, y un botón “Nueva imagen”, que envía un mensaje al sistema domótico para que este genere una nueva imagen, que sustituirá a la que aparecía en la pantalla.

La imagen está situada en el centro de la pantalla, de manera que es el elemento que recibe más atención por parte del usuario. La hemos rodeado de un borde blanco, para que no se generen confusiones entre la imagen generada, y el fondo de la aplicación.

3.2.3: Diagrama de navegación

Un diagrama de navegación, es una representación visual de las conexiones y jerarquías de las distintas pantallas dentro de una misma aplicación.

Dado que la aplicación está enfocada a todo tipo de usuarios, se ha intentado conseguir una navegabilidad de la aplicación muy sencilla, de forma que el usuario pueda reconocer rápidamente y sin necesidad de aprendizaje o memoria, donde tiene que pulsar para poder realizar la acción que desea llevar a cabo.

El diagrama de navegación que resulta de aplicar esto es el siguiente:

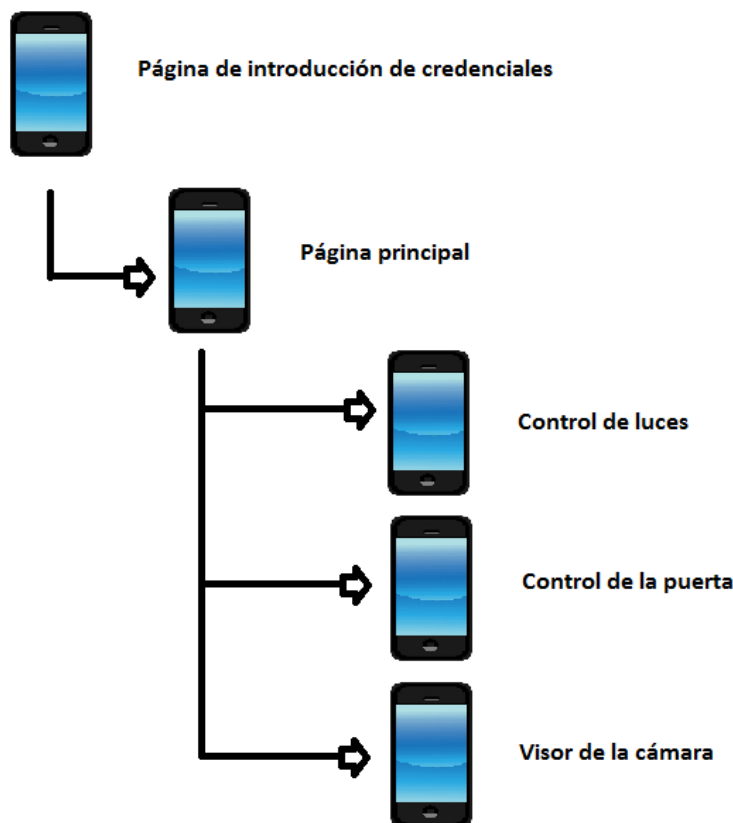


imagen 38: Diagrama de navegación

3.3 diseño hardware

en la siguiente imagen, se muestra un esquema del diseño hardware, y de cómo está conectado entre sí:

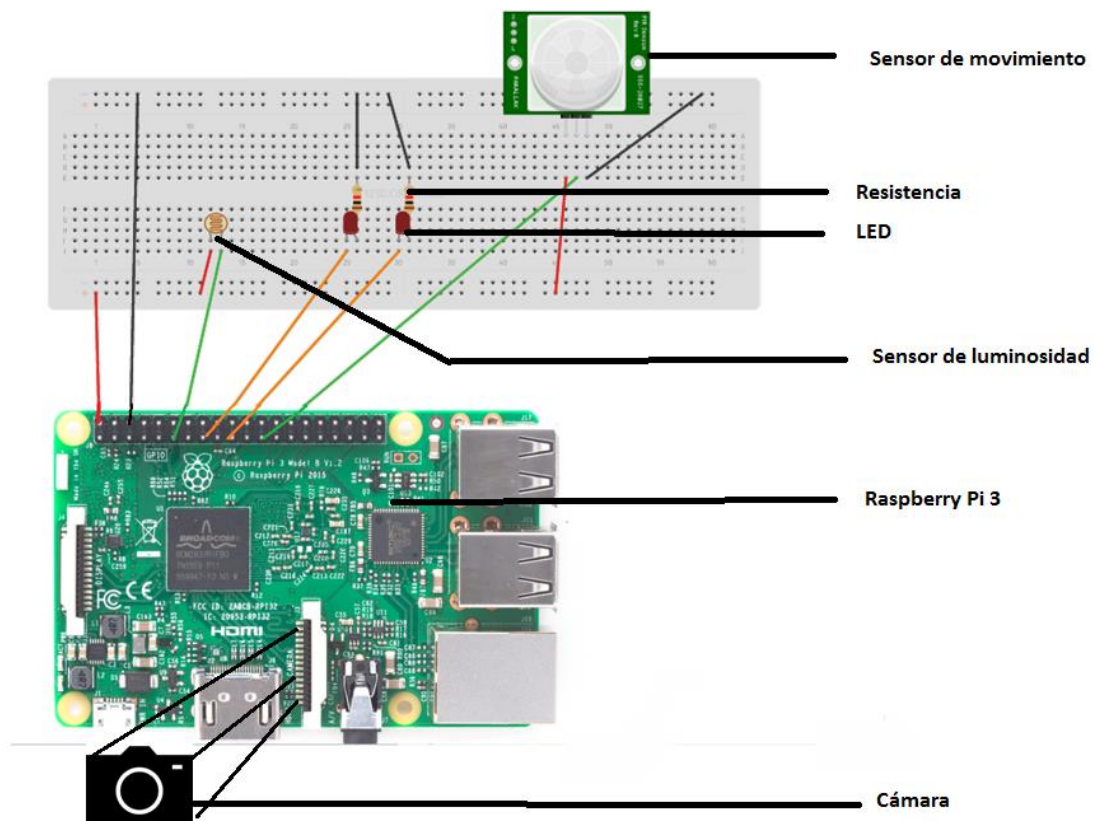


imagen 39: esquema diseño hardware

en la imagen, contamos con los elementos descritos en capítulos anteriores, un sensor de movimiento, dos resistencias, dos LED, una Raspberry Pi 3, un sensor de luminosidad o LDR, y una cámara.

En la imagen, los cables rojos representan conexiones con el voltaje de 5 voltios que aporta la raspberry, los cables grises son conexiones a tierra, los cables naranjas son representados salidas digitales (outputs), y los cables verdes entradas analógicas (inputs)

Si observamos las conexiones de la raspberry que aparece en la imagen, con la imagen de abajo que describe los pines de esta placa, podemos observar como el cable rojo está conectado a 5V, el cable gris a tierra, y el resto de cables están conectados a los pines descritos como GPIO, que son los pines utilizados para input/output.

Los LED están conectados en su salida a una resistencia cada uno para protegerlos de un posible exceso de voltaje, que podría provocar que los LED se fundieran.

La cámara de la raspberry pi, está representada con el icono de una cámara, pero, la conexión de la cámara real se realiza en el mismo punto que señala este icono.

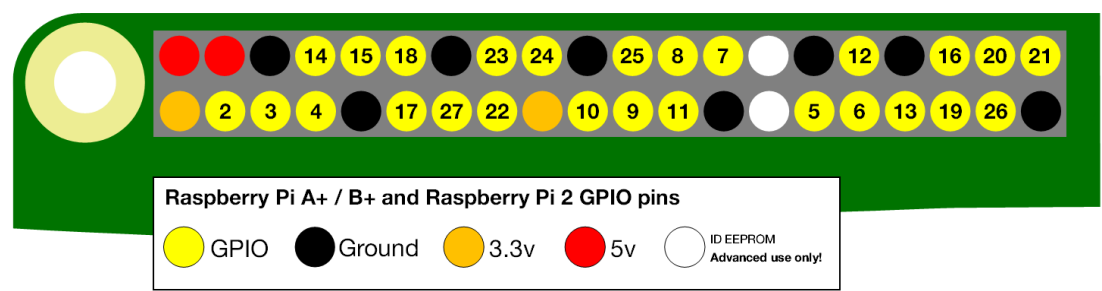


imagen 40: Distribución de pines de la Raspberry pi 3

Capítulo 4:

Implementación y pruebas

Como ya se ha mencionado en varias ocasiones, este proyecto trata sobre diseñar e implementar un sistema de control que simule el comportamiento de un sistema domótico en el entorno cercano a la puerta, controlado mediante una aplicación móvil

En este capítulo vamos a describir la implementación realizada en los diferentes módulos, para llegar a implementar el sistema completo. Además, incluirá un apartado en el que se describen los cálculos y procesos que implementa el sistema de tiempo real.

4.1 implementación

En este apartado, describiremos como se han implementado los dos sistemas principales que forman este proyecto.

4.1.1 implementación de la aplicación móvil.

La aplicación móvil, es donde el usuario ve reflejada la funcionalidad de la aplicación, y es la parte del sistema que incluye la interfaz gráfica, por ello, cada una de las pantallas incluye un código xml, y un código Java.

En el código xml es donde se refleja la interfaz gráfica que hemos diseñado anteriormente, utilizando la ayuda de Android Studio, que es el programa que hemos utilizado para implementar esta parte del sistema, es muy sencillo ir colocando los elementos que aparecen en el diseño en cada una de las pantallas, aunque de momento no tienen ninguna funcionalidad.

Para permitir que el código java reconozca los botones, cajas de texto, y demás elementos que añadimos en la interfaz, a cada uno de estos elementos le añadimos un campo en el código xml de la siguiente forma: `Android:id = "@+id/XX"`, donde XX es el id que asignamos a este elemento. Para facilitar la lectura de estos id, asignamos una id específica para cada tipo de elemento, es decir, si queremos dar un id al botón de id, este será `"btn_abrir"`, de esta manera no necesitaremos pensar a que corresponde cada id.

En el código java asignado a cada pantalla, es donde implementamos la funcionalidad de la aplicación. El código java, cuenta con una función que salta cada vez que se inicia la pantalla a la que está asociada, por lo que, en esta función, es donde vamos a declarar la mayoría de nuestra funcionalidad.

Lo primero que tenemos que hacer, es asignar variables a cada uno de los elementos que componen la pantalla que estamos tratando, lo hacemos utilizando la id que asignamos en el xml.

Dado que los botones son los que generan el funcionamiento, son los elementos en los que nos tenemos que centrar. Generamos para cada botón una alerta que salta cuando es pulsado, y, dentro de esta alerta incluimos la funcionalidad que tiene que activar cada botón.

Si la acción que tiene que realizar es un cambio de pantalla, el cambio de pantalla se realiza utilizando un `"intent"`, a este `"intent"` hay que indicarle cual es la pantalla actual y cuál es la pantalla siguiente. Antes de hacer efectivo el cambio de pantalla, añadimos al `"intent"`, la información de la aplicación utilizando extras. Esta información es la que se refiere al estado de las luces y la puerta, de manera que no se pierdan los cambios

realizados al cambiar de pantalla. Una vez tenemos la información añadida, hacemos efectivo el cambio.

En el caso de los botones que interactúan con el sistema domótico, estos tienen que enviar información a este sistema. Lo primero que hacemos, es modificar el estado de la propia aplicación móvil, si ponemos el caso de encender y apagar las luces, antes de actuar, comprueban si las luces están deshabilitadas.

Una vez el estado de la aplicación ha sido actualizado, se llama a la función de comunicación, y se le pasa una cadena de caracteres con el código de lo que quieres que realice el sistema domótico, estos códigos están formados por el elemento con el que se desea interactuar, seguido de la acción que se desea realizar, separados por una barra baja: "LI" indica las luces, "DO" indica la puerta y "CA" se refiere a la cámara, luego la acción que se desea realizar puede ser "On" si se desea encender o abrir, "Off" si se desea apagar o cerrar, "hab" para habilitar, "des" para deshabilitar y "new" para generar una nueva imagen. Si quisiéramos encender las luces, enviaríamos "LI_ON".

La función de comunicación se conecta a la dirección ip local a la que está conectada la raspberry, y envía por ssh un mensaje con el código que hemos establecido previamente.

En el caso de la cámara, la función de comunicación además recibirá la imagen que ha generado este sistema domótico, entonces, llamara a una función exclusiva de esta pantalla que es la encargada de almacenar la imagen en la memoria de la aplicación móvil, sustituyendo la que estaba almacenada previamente, en el caso de que la hubiera, con el mismo nombre de archivo, de manera que se cambia automáticamente la imagen que se ve en la pantalla.

4.1.2 implementación del sistema domótico.

El sistema domótico, es el encargado de recibir las ordenes de la aplicación móvil, en interactuar con los actuadores en consecuencia. Este sistema está desarrollado en C, y cuenta con varias funciones que detallaremos a continuación:

En el código, para cada elemento que compone el sistema domótico existe una función independiente, ya sea un sensor o actuador. Además, contamos con una función principal que es la que coordina todo, y una función exclusivamente para recibir y enviar las comunicaciones. La información recibida por cualquiera de estas funciones, se almacena en variables globales, de forma que, si otra función las necesitara, únicamente tenga que leer estas variables.

La función principal, o Main, es la que tiene implementado el algoritmo del sistema de tiempo real que será descrito en profundidad en el apartado siguiente. Esta función es la que va llamando a las demás en función de lo que establezca el algoritmo.

Las funciones de los sensores implementan lo que es conocido como un conversor analógico digital. Lo que realizan estas funciones es, cuando son invocadas, leen el

ultimo valor que ha generado el sensor al que están asociadas. Luego determinan si el valor está incluido en el rango de positivos o negativos, y modifican la variable global pertinente. En el caso del sensor de luz, si el valor recibido es menor del valor establecido como límite entre luz y oscuridad, establece que hay oscuridad, y modificaría la variable global LuzExterior a 0, indicando al resto de funciones que está oscuro en el exterior.

En el caso de los actuadores, si se ha recibido la orden pertinente, es decir, si la variable global de esa orden está a 1, el actuador tendrá que hacer su trabajo, en el caso de las luces y la puerta, tendrán que emitir la señal pertinente por el pin de la raspberry al que esté conectado dicho actuador.

En el caso de la luz exterior, cuenta además con el modo automático, para el cual tiene que comprobar que las luces estén habilitadas, y que los sensores hayan dado los valores correctos para que se active, en cuyo caso, encenderá las luces emitiendo la señal por el pin.

En el caso de la función de la cámara, la cámara de la raspberry tiene que asociarse con el código de la siguiente manera: `camara = picamera.PiCamera()`, para tomar la imagen, ejecutamos la siguiente instrucción `camara.capture("imagen.jpeg")`. La imagen se almacena en el directorio de la raspberry que se especifique entre paréntesis, en este caso, se almacenaría en el mismo directorio en el que este el código que se está ejecutando, con el nombre "imagen.jpeg". siempre se almacenan en el mismo directorio y con el mismo nombre para que la función de comunicación pueda acceder a ella fácilmente.

La función de comunicación, lee de un buffer asociado a su dirección ip, que es donde se almacenan los mensajes recibidos desde la aplicación móvil. Cuando recibe una orden, la interpreta y modifica la variable global asociada a dicha orden en consecuencia.

4.2 Sistema de tiempo real

El sistema se basa en la captación de información desde unos sensores, así como de peticiones de lectura y escritura desde la aplicación móvil controlada por el usuario. El sistema controlara y analizara toda esta información para utilizar en la manera debida cada uno de los instrumentos conectados al sistema para simular el entorno real. De esta manera, podemos generar un esquema que imita el sistema que utilizaremos durante el desarrollo:

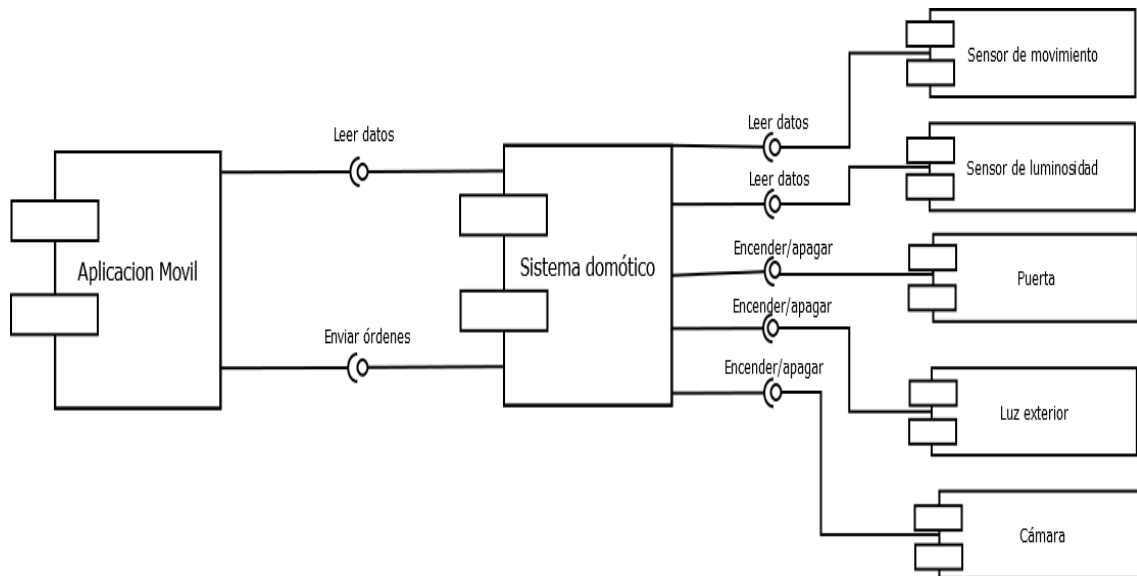


imagen 41: componentes que intervienen en el sistema

Las tareas a controlar por el sistema domótico son las siguientes:

- Encendido/apagado de la luz exterior.
- Apertura/cierre de la puerta.
- Tomar imagen con la cámara.
- Lectura del sensor de movimiento.
- Lectura del sensor de I
- Comunicación con la aplicación móvil
 - Lectura:
 - Recibir imagen
 - Peticiones:
 - Encendido/apagado de la luz exterior.
 - Apertura/cierre de la puerta.
 - Tomar imagen con la cámara.

Los detalles de las tareas son los siguientes:

Encendido/apagado de las luces.

Esta tarea, simula mediante un led conectado al sistema, la iluminación de una luz exterior, si hay alguien en la puerta, y está oscuro, la luz se enciende, si esa persona se aleja de la puerta, se apaga.

La tarea se lanza cuando el sensor de movimiento detecta movimiento, lo primero se comprueba si las luces están habilitadas, entonces se comprueba si el sensor de luz

detecta si está oscuro en el exterior. Si es así, la luz se enciende, cuando el sensor deja de detectar movimiento, si la luz está encendida, se apaga.

La tarea también se activa si el sistema recibe el mensaje *LG_XX* donde *XX* será *ON* para encenderlas, y *OF* para apagarlas, la tarea sigue los mismos pasos que en el caso anterior.

Apertura/cierre de la puerta.

Esta tarea simula, mediante un led conectado al sistema, la apertura o cierre de la puerta, de manera que, si el led está encendido, la puerta estaría abierta, en el caso de que este apagado, cerrada.

La tarea se activa cuando el sistema recibe un mensaje de la aplicación móvil de la forma *DO_XX*, donde *XX* puede ser *OP* si se desea abrir la puerta, y *CL* en el caso de cerrarla.

Tomar imagen con la cámara

Esta tarea simula la acción de una cámara de seguridad situada en el exterior de la puerta, mediante una cámara conectada al sistema.

La tarea se activa de dos maneras, la primera cuando el sensor de movimiento se activa después de un periodo de inactividad, la otra es cuando el sistema recibe un mensaje de la aplicación móvil de la forma *TK_IM*.

En ambos casos, el sistema le da la orden a la cámara para que tome la imagen, y después manda está a la aplicación móvil para que la muestre.

Lectura del sensor de movimiento.

Esta tarea es la encargada de controlar el sensor de movimiento conectado al sistema de manera que simula la acción que ejercería un sensor conectado a la puerta para saber si hay alguien cerca.

La tarea se activa cada 500 milisegundos, ya que está establecido en los requisitos. Para almacenar la información, cuenta con dos variables globales, una en la que establece si se ha detectado movimiento en su última lectura, y otra para almacenar lo detectado en la anterior. De esta manera, se puede saber si la persona acaba de llegar, o simplemente está en el entorno de la puerta.

Lectura del sensor de luz.

Esta tarea es la encargada de detectar si en el exterior hay suficiente luz o no, de manera que el sistema pueda conocer cuándo debe encender la luz exterior.

La tarea se lanza cada 500 milisegundos, tal y como esta especificado en los requisitos, y recoge en una variable global si en la última lectura el sensor ha detectado si hay luz en el exterior. El valor intermedio para determinar si hay luz, será 200.

Comunicación con la aplicación móvil

Estas tareas cuentan con los siguientes periodos y tiempos de computo:

Tarea	Periodo(s)	Tiempo de cómputo(es)
Encendido/apagado de la luz exterior	20	0.60
Apertura/cierre de la puerta.	20	0.60
Tomar imagen con la cámara	10	0.60
Lectura del sensor de movimiento	5	0.60
Lectura del sensor de luz	20	0.60
Comunicación con la aplicación móvil	10	0.75

Tabla 66: periodos y tiempos de computo

Planificador cíclico

Una vez hemos establecido todas las tareas, los periodos y tiempo de cómputo de las mismas, debemos elaborar un planificador cíclico que satisfaga dichas restricciones, para ello lo primero es saber si es posible su planificación, para ellos calcularemos la **utilización de CPU** de las tareas, que se calcula de este modo:

$$U = \sum_{i=1}^n \left(\frac{C_i}{T_i} \right)$$

Siendo n el numero de tareas

En el caso que nos ocupa la utilización de la CPU es: 0,29

Sabiendo que las tareas son potencialmente panificables hay que proceder a calcular el **hiperperiodo** del mismo, de este modo: $T_m = m.c.m(T)$; *Siendo T el conjunto de peridos de la tareas.*

El hiperperiodo en este caso sería 30 s, después, debemos calcular el periodo secundario, que seguiría estas condiciones:

- Debe ser mayor que todos los tiempos de cómputo y menor que todos los periodos.

$$\max(C) \leq Ts \leq \min(T)$$

- Debe ser múltiplo de alguno de los periodos. $\exists i=1n TsTi=k, k \in \mathbb{N}$
- Debe ser divisor exacto del hiperperiodo. $TmTs=k', k' \in \mathbb{N}$
- Debe cumplirse que: $\forall i=1n (Ts + (Ts - m.c.d.(Ts, Ti)) \leq Ti)$

Para cumplir todas estas condiciones hemos elegido un ciclo secundario de 250ms. Con un ciclo secundario de 5 ms el ciclo principal contiene 6 ciclos secundarios, ahora debemos acomodar las tareas en ellos. Para ello, una tarea sólo podrá ejecutarse en un ciclo secundario sea completamente válido para ella, es decir, que en ningún momento del mismo haya pasado más tiempo que el periodo de la tarea desde su última ejecución. Los ciclos secundarios quedan como sigue:

Tiempo (s)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Luz exterior																														
Puerta																														
Cámara																														
movimiento																														
Sensor de luz																														
Comunicación																														

Tabla 67: ciclos secundarios de las tareas.

Una configuración válida para estos ciclos secundarios sería:

Tiempo (s)	1	2	3	4	5	6	7	8	9	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	3	
										0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
Luz exterior																														
Puerta																														
Cámara																														
movimiento																														
Sensor de luz																														
Comunicación																														

Tabla 68: configuración válida de los ciclos secundarios.

Implementación del planificador

Para la implementación del planificador se ha usado la siguiente configuración:

- Inicio del planificador
 - Si($\text{mod}(\text{contador}, \frac{T_m}{T_s})$)
 - Es 0
 - Comprueba el sensor de movimiento.
 - Decide si tomar la fotografía.
 - Escucha las ordenes de la app.
 - Es 1
 - Decide si encender la luz exterior.
 - Decide si abrir la puerta.
 - Comprueba el sensor de movimiento.
 - Mide la luz exterior.
 - Es 2
 - Comprueba el sensor de movimiento.
 - Decide si tomar la fotografía.
 - Escucha las ordenes de la app.
 - Es 3
 - Decide si encender la luz exterior.
 - Decide si abrir la puerta.
 - Comprueba el sensor de movimiento.
 - Mide la luz exterior.
 - Es 4
 - Comprueba el sensor de movimiento.
 - Decide si tomar la fotografía.
 - Escucha las ordenes de la app.
 - Es 5
 - Comprueba el sensor de movimiento.
 - End si
 - $\text{Tiempo1} = \text{contador} \times T_s$;
 - $\text{Tiempo2} = \text{tiempoActual}$;
 - Esperar ($\text{tiempo2} - \text{tiempo1}$);
 - $\text{Contador} = \text{contador} + 1$;

- Fin del planificador

Después de realizar las tareas que correspondan al ciclo secundario en el que se esté hay que esperar hasta completar todo el ciclo secundario que se está ejecutando, por ejemplo, si en uno de estos ciclos que dura 5 s se acabaran las tareas en 2, habría que esperar 3 para volver al siguiente ciclo. Esto se hace así para que el modelo sea regular en el tiempo y se mantengan sus propiedades.

4.3 Pruebas

Una vez la aplicación esta implementada, hay que proceder a efectuar una de las partes más importantes del desarrollo, y consiste en probar que la aplicación funcione correctamente y no tenga comportamientos imprevistos o aleatorios. Para ello, comprobamos que cada módulo funciona correctamente

Las pruebas que mostraremos a continuación, se han realizado en un entorno controlado donde se comprueban las respuestas que aporta el sistema a una serie de acciones. La respuesta del sistema debería de coincidir con lo especificado en los requisitos.

Las tablas cuentan con los siguientes campos:

- **ID:** identificador univoco de la prueba, sigue la siguiente nomenclatura PR_XX donde XX es un numero incremental empezando por 01.
- **Descripción:** describe el objetivo que busca probar la prueba en cuestión.
- **Resultado esperado:** reacción esperada del sistema, puede ser “fallo”, en caso de que el sistema deba dar un error, o “acierto” si el sistema debe funcionar correctamente.
- **Resultado obtenido:** indica el resultado devuelto por el sistema una vez se han ejecutado los pasos, puede tomar los valores “fallo” o “acierto”.
- **Ha pasado la prueba:** indica el sistema ha superado la prueba, puede ser “correcto” en el caso de que la haya pasado, o “incorrecto” en caso contrario.
- **Acciones:** pasos a seguir para realizar la prueba.

ID	PR_01
Descripción	Introducir un usuario incorrecto y una contraseña correcta
Resultado esperado	Fallo
Resultado obtenido	Fallo
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la aplicación • Introducir un usuario incorrecto • Introducir la contraseña válida

Tabla 69: PR_01

ID	PR_02
Descripción	Introducir un usuario incorrecto y una contraseña incorrecta
Resultado esperado	fallo
Resultado obtenido	fallo
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la aplicación • Introducir un usuario incorrecto • Introducir la contraseña no válida

Tabla 70: PR_02

ID	PR_03
Descripción	Introducir un usuario correcto y una contraseña incorrecta
Resultado esperado	Fallo
Resultado obtenido	Fallo
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la aplicación • Introducir un usuario correcto • Introducir la contraseña no válida

Tabla 71: PR_03

ID	PR_04
Descripción	Introducir un usuario correcto y una contraseña correcta
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la aplicación • Introducir un usuario correcto • Introducir la contraseña válida

Tabla 72: PR_04

ID	PR_05
Descripción	Encender las luces estando habilitadas
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de luces • Comprobar que las luces aparecen habilitadas • Pulsar en apagar

Tabla 73: PR_05

ID	PR_06
Descripción	Apagar las luces estando habilitadas
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de luces • Comprobar que las luces aparecen habilitadas • Pulsar en apagar

Tabla 74: PR_06

ID	PR_07
Descripción	Deshabilitar las luces estando encendidas
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de luces • Comprobar que las luces aparecen habilitadas • Encender las luces • Pulsar en deshabilitar las luces

Tabla 75: PR_07

ID	PR_08
Descripción	Deshabilitar las luces estando apagadas
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de luces • Comprobar que las luces aparecen habilitadas • apagar las luces • Pulsar en deshabilitar las luces

Tabla 76: PR_08

ID	PR_09
Descripción	Habilitar las luces
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de luces • Comprobar que las luces aparecen deshabilitadas • Pulsar en habilitar las luces

Tabla 77: PR_09

ID	PR_10
Descripción	Apagar las luces estando deshabilitadas
Resultado esperado	Fallo
Resultado obtenido	Fallo
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de luces • Comprobar que las luces aparecen deshabilitadas • Pulsar en apagar

Tabla 78: PR_10

ID	PR_11
Descripción	Encender las luces estando deshabilitadas
Resultado esperado	Fallo
Resultado obtenido	Fallo
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de luces • Comprobar que las luces aparecen deshabilitadas • Pulsar en encender

Tabla 79: PR_11

ID	PR_12
Descripción	Abrir la puerta
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de la puerta • Pulsar en abrir

Tabla 80: PR_12

ID	PR_13
Descripción	Cerrar la puerta
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de la puerta • Pulsar en cerrar

Tabla 81: PR_13

ID	PR_14
Descripción	Observar la imagen de la cámara
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de la cámara • Observar que aparece la imagen

Tabla 82: PR_14

ID	PR_15
Descripción	Generar una nueva imagen
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Entrar en la pantalla de la cámara • Observar que aparece la imagen • Pulsar en “nueva imagen” • Observar que aparece la nueva imagen

Tabla 83: PR_15

ID	PR_16
Descripción	Comprobar el encendido automático de luces de día
Resultado esperado	Fallo
Resultado obtenido	Fallo
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Comprobar que es de día • Activar el sensor de movimiento • Comprobar que no se encienden las luces

Tabla 84: PR_16

ID	PR_17
Descripción	Comprobar el encendido automático de luces de noche
Resultado esperado	Acierto
Resultado obtenido	Acierto
Ha pasado la prueba	Correcto
Acciones	<ul style="list-style-type: none"> • Comprobar que es de noche • Activar el sensor de movimiento • Comprobar que se encienden las luces

Tabla 85: PR_17

Capítulo 5: Planificación y presupuesto

En este capítulo, se añadirá la planificación que se realizó para realizar el desarrollo del proyecto, desglosado en tareas y roles utilizados, a su vez, se explicaran y mostraran con detalle todos los recursos, ya sea tiempo, dinero u otros elementos, invertidos en el mismo, desglosados para cada una de sus fases

5.1 Planificación

En este apartado, vamos a detallar la planificación utilizada durante la realización del proyecto.

En la siguiente tabla, viene detallado para cada una de las tareas en las que se divide el proyecto, su fecha de inicio, la de fin, y el lapso total de días requerido para realizar dicha tarea. Las fechas tendrán el siguiente formato dd/mm/yyyy.

Las fiestas nacionales se han excluido de la planificación.

Tarea	Fecha de inicio	Fecha de Fin	Duración (Días)
Estudio de la viabilidad	21/03/2017	25/03/2017	4
Documentación inicial	26/03/2017	31/03/2017	5
Análisis	01/04/2017	06/04/2017	5
Diseño	07/04/2017	15/04/2017	8
Implementación	16/04/2017	06/05/2017	20
Pruebas	07/05/2017	17/05/2017	10
Documentación	18/05/2017	10/06/2017	23

Tabla 86: planificación del proyecto

5.1.1 Diagrama de Gantt

Las actividades plasmadas en la tabla anterior, también pueden ser plasmadas utilizando un diagrama de Gantt, como el siguiente:

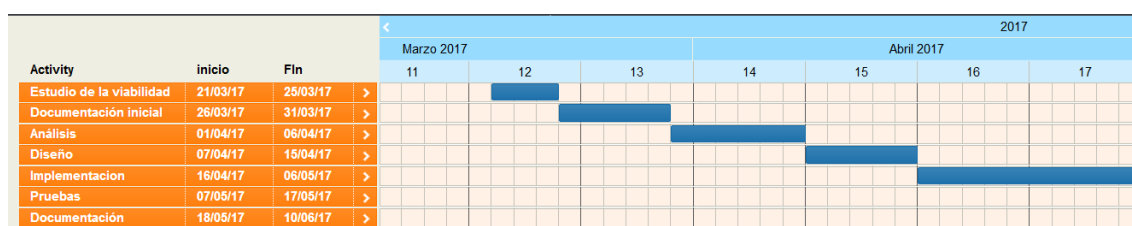


imagen 42: Diagrama de Gantt [1/2]

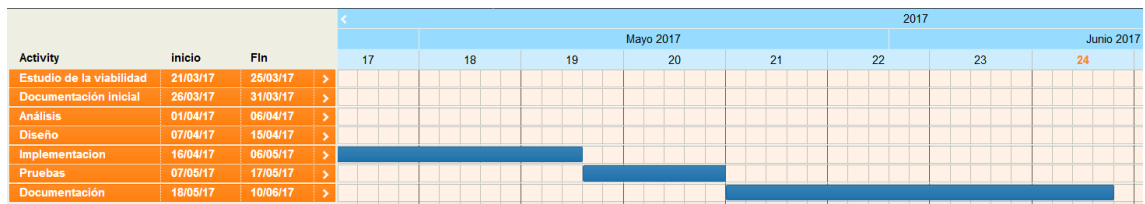


imagen 43: Diagrama de Gantt [2/2]

5.2 Presupuesto

En este apartado, se detallarán todos los recursos invertidos durante el desarrollo del proyecto, y los costes de los mismos desglosados en función de su naturaleza.

Los costes se detallarán con el coste bruto, la amortización que se aplica a cada elemento, y los impuestos que se le aplican. Además, se incluirán los cálculos que se han realizado para obtener estas cifras.

5.2.1 tiempo dedicado

A continuación, mostraremos detalladamente las horas totales que ha requerido el desarrollo del mismo:

- **Estudio de viabilidad:** 4 días x 6 horas = 24 horas.
- **Documentación inicial:** 5 días x 6 horas = 30 horas.
- **Fase de Análisis:** 5 días x 6 horas = 30 horas.
- **Fase de Diseño:** 8 días x 6 horas = 48 horas.
- **Fase de Implementación:** 20 días x 6 horas = 120 horas.
- **Fase de Pruebas:** 10 días x 6 horas = 60 horas.
- **Documentación final:** 23 días x 6 horas = 138 horas.

Las horas totales dedicadas al proyecto son 6 horas diarias durante 75 días, o lo que es lo mismo 450 horas.

5.2.2 Coste personal

En este apartado, calcularemos el coste del personal implicado en el desarrollo del proyecto, en función del rol que ha desempeñado. La siguiente tabla muestra el detalle de los costes:

Nombre Completo	Rol desempeñado	Coste (Hombre/mes)	Coste total
Imanol Sanz Ruiz	Diseñador	2.500	5.100
Imanol Sanz Ruiz	Analista	2.100	4.250
Imanol Sanz Ruiz	Programador	1.500	2.200
Imanol Sanz Ruiz	Tester	1.000	1.800
Total			13.350

Tabla 87: costes de personal

* todos los costes estarán representados en euros.

El coste total ha sido calculado aplicando la siguiente ecuación:

$$\text{Coste} = \text{total días} * \text{horas/día} * \text{Coste hombre/mes}$$

5.2.3 Coste de hardware

Para el desarrollo del proyecto, ha sido necesaria la adquisición de un terminal

portátil que servirá para el desarrollo de cada una de las fases del proyecto, además, para el sistema domótico, se ha adquirido una raspberry pi 3, con sus sensores y complementos necesarios para el proyecto.

A continuación, se detallan los datos de los dispositivos, para el cual se ha calculado el coste imputable a partir de la fórmula de amortización siguiente:

$$\frac{A * C}{B}$$

Donde:

- A -> meses de utilización del dispositivo para el proyecto.
- B -> periodo de amortización
- C -> Coste del dispositivo

Descripción	Coste	Meses de uso	Amortización	Coste imputable
Portátil Acer Aspire VX15 15,6" 2.8 GHz 16 Gb RAM 500 Gb Disco Duro	1099.00	4	36	274,75
Raspberry pi 3	30.00	4	12	10.00
Sensores	10.00	-	-	10.00
Cámara raspberry	25.00	-	-	25.00
			TOTAL	319,75

Tabla 88: Costes de hardware

* todos los costes estarán representados en euros.

** los sensores y actuadores no cuentan con periodo de amortización.

5.2.4 Coste de software

En esta sección, vamos a detallar el coste de los programas utilizados durante el desarrollo, y que requerían de la compra de una licencia, ya que no eran de código libre, y eran necesarios para el desarrollo de ciertas fases del proyecto.

Descripción	Coste
Microsoft Office Professional 2013	499
Adobe Photoshop CS5	250
Desarrollador Google Play	25
Total	774

Tabla 89: Costes de Software

* todos los costes estarán representados en euros.

5.2.5 Costes totales

Una vez hemos mostrado el detalle de todos los costes asociados al proyecto, procedemos a mostrar el total del coste del proyecto. Para este total tendremos en cuenta los costes calculados en los apartados anteriores, a los que añadiremos un 20% de costes indirectos que cubrirán los posibles riesgos y gastos imprevistos que surjan.

Descripción	Coste total
Personal	13.350,00
Hardware	319,75
Software	774,00
Coste indirecto (20%)	2.888,75
Total, sin I.V.A	17.332,50
Total, con I.V.A (21%)	20.972,32

Tabla 90: Costes totales

* todos los costes estarán representados en euros.

Chapter 6: Conclusions and future lines

In the current chapter are exposed the conclusions reached with this project development. There are also exposed the future lines of work that will be followed in order to improve and complete the system we are currently presenting.

6.1 Conclusions

Once we have finished the project, it is necessary to carry out a general study of what the project has supposed, goals that have been achieved and objectives surpassed during the development process.

We will discuss everything listed above in two parts, one for the project, and other for the personal conclusions.

6.1.1 Project conclusions

The main objective of the project is to create a domotic system with the ability to be controlled by a mobile application. The domotic system had to comply with maximum response times in all cases to avoid the user having to wait too long for a response.

The main idea that has been made using the prototype we have developed has been the correct communication between the two parts that make up our system, managing to solve the main problem that gave us the idea to develop this system.

It should be noted that the objectives set at the beginning of the project have been successfully achieved:

- A controlled home automation system that controls the sensors and actuators has been successfully implemented.
- The application has been developed for the user, creating a friendly interface that displays the data.
- The two parts of the system have been successfully communicated, avoiding errors.

This has only been possible thanks to having clear the established requirements, and the interfaces of communication between the different parts, fact that has allowed us to implement the home automation system and the mobile application separately, and that do not generate problems when putting them together.

Thanks to these requirements, it has also been possible to achieve the sub-objectives set out at the beginning of the project:

- An interface for the mobile application has been designed friendly and does not need the use of memory of the user.
- The user's personal data has been stored correctly encrypted.
- A real-time system to control the home system has been successfully implemented.
- High error tolerance has been achieved throughout the system.

During the development of the project we have followed the planning that we have presented in this document to be able to fulfill all the objectives set. With all this we can ensure that the objectives set for this project have been satisfactorily achieved.

6.1.2 Personal conclusions

This project was a personal challenge for me, ever since i started the computer technology bachelors degree, i have loved developing software for electronic devices, i didnt care what the electronic device was, a remote controlled car, some LED related Project, or other crazy things that are out there. So, when i had the chance to develop a remote controlled home automation system, which could be used in real life, i couldnt say no to it.

I didnt want to make a normal home automation system which could only control some ligths and windows, i wanted to make a system that could solve or help with a real problem our users may suffer.

After some thought, i realised that one of the most annoying things i had endured related with my house, was when i forgot my keys inside it, and there was noone inside, and i decided to try solving this annoyance.

At this point we have reached all of the points we wanted to include in this project, we made the user friendly Android application, we made the home automation system, and we successfully conected both of this modules for their communication.

The main conclusión reached at this point of the development of the project is that it was a lot harder than it seemed at the beggining. It has required a lot more time and effort than it was planned at first.

Even thoug, we didnt need to learn how to use any of the technology used in this Project, we still needed a lot of wok to communicate the home automation system

with the application, the other two parts weren't as difficult. The Android application was the easiest part of all, because it doesn't have a complex structure, it only required some time to thought and design the user interface which it was going to use. Talking about the home automation system, we did a previous project very similar to this one, so, the bigger problem we faced implementing this module was the mathematic part related to it, which makes sure the project is viable and all the times collected at the requisites will be respected.

The hardware needed to create a proper model of the system, was too expensive and hard to use. We used a Raspberry Pi for this Project instead of an Arduino, just in case we could add some of this hardware, because the Arduino couldn't use this hardware. We thought about using some servomotors for the door opening simulation, but they aren't so cheap, and require some complex calculations and an external power source, so we refused using them at the end.

We even looked for a real electronic lock, but it was refused a lot faster than the servomotors, due to the high price of these machines, and the complexity of the software to control them.

6.2 Future working lines.

The final Project we have submitted, is far from being a real full commercial product, but, as we said in the conclusion section, we had some complexity and time problems, so we couldn't include all the things we wanted to include at first.

There are lots of things we wanted to improve about the Android mobile application, first would be to include a new screen, or pop up, linked to the main screen, from which the user could change some of the application settings, such as the user id, the password, or the language in which the application is shown.

We would like to include a database, probably the SQLite, which has full compatibility with Android. We would use this DB for storing all the user related data, that we don't want to have stored directly at the application code, and we would like to store this data ciphered, so it would be even harder to get it for an attacker.

The communication system is the biggest security hole in most of the modern applications, so we would like to make ours a bit safer, using a handshake protocol between the application and the home automation system. This protocol makes the two parts of the communication to share and develop some keys, before they start to send the real data, so when they send it, the transmitter could cipher this data with the keys, and the receiver could decipher it.

The home automation system growth possibilities has no limits, so we would like to add some more elements to it, in order to make a more useful system, and we would add the appropriate screen to the mobile application. We would really like to replace the current model we have for displaying the system functionality, for the real hardware parts, so we could show how this project really works, and it would also help selling the final product, if we feel like selling it.



imagen 44: electronic lock example

The real system element that we would most like to count on to be able to use, and that would really prove that our project works, and that can be used in the real world, once all the security that we mentioned previously was included, would be the electronic lock, since it is the key element around which this project is developed.

Anexo 1: Glosario

- **ID:** muestra el identificador del requisito en cuestión.
- **RU:** requisito de usuario.
- **Información sensible:** información personal del usuario que no se debe hacer publica
- **Firewall:** Dispositivo software de seguridad que controla el tráfico de entrada y salida a la red de acuerdo con una política de control de acceso
- **Framework:** Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado
- **GB:** gigabyte
- **GHz:** gigahercio
- **Http:** Hyper Text Tranfer Protocol
- **IDE:** Integrated Development Enviroment
- **IVA:** impuesto sobre el valor añadido
- **JDK:** Java development kit
- **WI-FI:** Wireless fidelity
- **DB:** base de datos

Anexo 2: Bibliografía

[1] Página de Android

<https://www.android.com/>

[2] Página de IOS

<https://www.apple.com/es/ios/ios-10/>

[3] Página de raspberry

<https://www.raspberrypi.org/>

[4] características del sistema operativo Android

[https://en.wikipedia.org/wiki/Android_\(operating_system\)](https://en.wikipedia.org/wiki/Android_(operating_system))

[5] características del sistema operativo IOS

<http://en.wikipedia.org/wiki/IOS>

[6] uso mundial de la telefonía móvil

<http://www.elperiodico.com/es/noticias/economia/los-usuarios-movil-superaran-los-5000-millones-2017-segun-gsma-5863401>

[7] mobile worl congress 2017

http://elpais.com/eventos/2017/02/24/mwc/1487900380_692466.html

[8] Definicion y características de los dispositivos móviles

https://es.wikipedia.org/wiki/Tel%C3%A9fono_m%C3%B3vil

[9] Smart home applications security

http://web.eecs.umich.edu/~earlence/assets/papers/smartthings_sp16.pdf

[10] aplicaciones domóticas más utilizadas

<http://www.independent.co.uk/extras/indybest/gadgets-tech/best-smart-home-devices-security-system-hub-camera-tech-amazon-echo-google-home-nest-a6800731.html>